AFRL-HE-BR-TR-2003-0120

UNITED STATES AIR FORCE
RESEARCH LABORATORY

LASER RANGE SAFETY TOOL (LRST)

PHYSICS REFERENCE

Gregg Crockett

Logicon-RDA
2600 Yale Blvd SE
Albuquerque, NM 87106

Boeing-SVS
4411 The 25 Way NE Ste 350
Albuquerque, NM 87109

HUMAN EFFECTIVENESS DIRECTORATE
DIRECTED ENERGY BIOEFFECTS DIVISION
OPTICAL RADIATION BRANCH
2650 LOUIS BAUER DRIVE
Brooks City-Base, TX 78235

September 2003

20031017 156

NOTICES

This report is published in the interest of scientific and technical information exchange and does not constitute approval or disapproval of its ideas or findings.

This report is published as received and has not been edited by the publication staff of the Air Force Research Laboratory.

Using Government drawings, specifications, or other data included in this document for any purpose other than Government-related procurement does not in any way obligate the US Government. The fact that the Government formulated or supplied the drawings, specifications, or other data, does not license the holder or any other person or corporation, or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

The Office of Public Affairs has reviewed this paper, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This report has been reviewed and is approved for publication.

Cecilia D. Mde Oca

CECILIA MONTES DE OCA, Capt, USAF
Contract Monitor

Miller

RICHARD L. MILLER, Ph.D.
Chief, Directed Energy Bioeffects Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
14-05-2003	Technical Report	1998-2001

4. TITLE AND SUBTITLE

Laser Range Safety Tool (LRST), Physics Reference Manual

5a. CONTRACT NUMBER
F04701-98D-100, CDRL A004

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Gregg Crockett

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Logicon-RDA
2600 Yale Blvd SE
Albuquerque, NM 87106
(1998-2001)

Boeing-SVS
4411 The25 Way NE Suite 350
Albuquerque, NM 87109
(2001-present)

8. PERFORMING ORGANIZATION REPORT

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Human Effectiveness Directorate, Directed Energy Bioeffects Division
Optical Radiation Branch
2650 Louis Bauer Drive
Brooks City-Base, TX 78235-5215

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/HEDO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-HE-BR-TR-2003-

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Technical Monitor: Capt. Cecelia Montes De Oca (210) 536-3035

14. ABSTRACT
LRST is a software tool for calculating reflected energy hazards in the context of test range development of the Airborne Laser (ABL) system. The LRST software package consists of a suite of programs that provide facilities for describing and simulating test scenarios, including the range, the ABL aircraft laser complement, the target, and the observers. This manual provides a detailed description of the physical basis of the simulation and the computational algorithms utilized to calculate the radiometry and hazards.

15. SUBJECT TERMS
Airborne Laser, laser safety, directed energy bioeffects, laser range safety tool, LRST

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UL	124	Lt Col William P. Roach
Unclass	Unclass	Unclass			19b. TELEPHONE NUMBER *(include area code)* (210) 536-4817

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

1.0 Physics Modeling Overview

The physics modeling actually starts by examining the basis for hazards, found in the ANSI Standard Z136. Unfortunately, the standard does not really clearly address the requirements for reflected laser energy, where the observed intensities can be rapidly varying functions of time. We can, however, deduce from their provided equations, how to fit our radiometric problem into the ANSI paradigm to make reasonable assessments of eye hazards. The underlying requirement of the LRST code will be to generate an accurate time history of intensity at potential positions where humans might be exposed to reflected laser light from a target missile being irradiated by an Airborne Laser (ABL).

There are about eight basic algorithmic steps necessary to evaluate eye hazards in LRST. Figure 1.1 shows a simple diagram of this process.
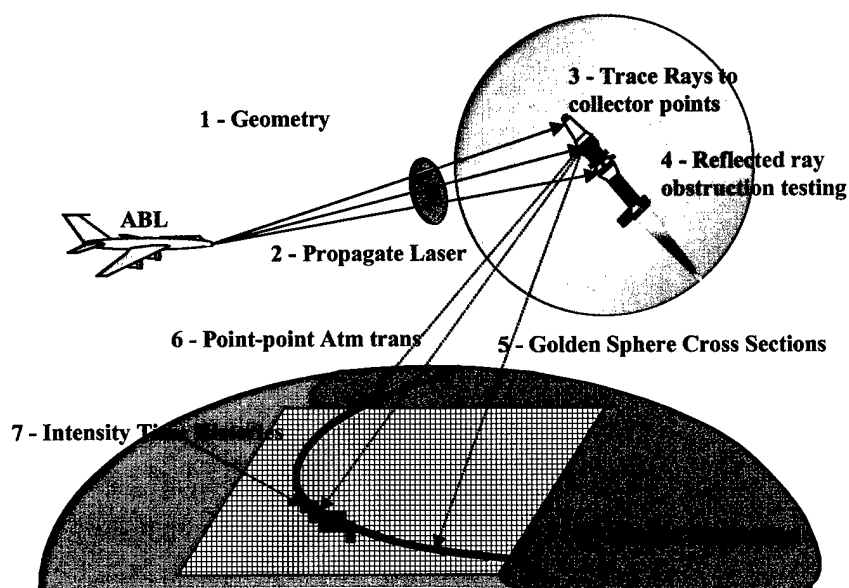
Figure 1.1. Eight Basic Algorithmic Steps to Evaluate Hazards

Before anything, the scenario geometry must be calculated. The position and velocity of the ABL is calculated. The ABL flies a great circle with a constant heading during the entire flight time of the missile. However, the body coordinate frame of the airplane has its z-axis along the local zenith, and the x-axis is along the velocity vector. A telescope frame is inserted between the aircraft frame and the missile to exactly point the ABL lasers at the missile.

The missile position and velocity is derived from a look-up table of pre-simulated position coordinates imported into LRST. The missile body frame points along the instantaneous velocity vector.

Centered on the missile position is a cross section sphere (referred to as the Golden Sphere) frame, which is a local-horizontal, local-vertical frame where the z-axis points along the local

zenith, the x-axis points northward, and y forms a right handed triad. The orientation of this frame necessarily changes at each time step.

The surface of the earth out to the horizon of the target is meshed with ground observers, which of course are conformal to the surface of the local terrain. In addition to ground observers, there are base boundary fences, and air corridor surfaces. Dynamic observers, such as we might find in chase planes, are located according to their positions and velocities.

The baseline geometry for LRST assumes typical theater ballistic missile (TBM) flight altitudes and speeds, and the ABL is put at about 40,000 feet or higher. In general, we assume all objects in the scenario are separated by at least 1 km to satisfy far field assumptions in the radiometry. We also assumed the missile was at least 10 km above the mean sea level to clear any cloud layer. While we would like to have all scenario objects positioned exactly, this is not possible since this information is never known. We can only estimate the actual ABL and the missile flight paths. Users will therefore need to make some parametric runs on missile fly-outs, or ABL flight paths to ensure there are no unfavorable geometries. Both the scenario geometry and scatter radiometry will change with time, so we will break all calculations down into finite time steps. The size of the time step should be small compared to the scene dynamics, and large enough to minimize redundant calculations. Typically this is on the order of one second.

Now we need to propagate the ABL lasers to the missile target plane. We currently model the laser irradiance at the missile with either a Gaussian profile or a uniform flood beam. The flood beam uses the peak irradiance of a diffraction limited Gaussian beam to uniformly cover the target. This is certainly a conservative approach that eliminates any worry about exact beam profile approximations or pointing errors. In essence the beam peak hits everywhere equally. This approach does create about ten times the initial beam energy on target however.

The Gaussian beam profile is more realistic, but must be pointed at a representative aimpoint. We assume that the beam is quasi-collimated at the target plane. Thus all the incident rays are parallel to each other to within a few micro-radians.

At the target, we must handle the radiometric scattering physics. We want the radiometrics to be as accurate as possible, while being conservative if any approximations are used. We start by constructing a target model that approximates the actual geometrical shape as closely as possible. We adopted the modeling paradigm used by the AFRL Satellite Assessment Center. They use a computer aided design (CAD) tool called the Solids Modeling Tool (SMT). It uses analytic quadrics such as cylinders and cones for primitive surface shapes. Since SMT is based on constructive solid geometry (CSG) operators, all primitives are formed into convex solids by placing planar end-caps on the quadrics where necessary. The classical CSG operators are Union (addition), Subtraction (voiding), and Intersection (overlap). Since most missiles are simple constructs of cylinders, cones, and plates (fins), this geometric modeling works fairly well. We have also adopted their supporting material properties database format (i.e., MATTER.DAT). There are other paradigms for doing three-dimensional CAD modeling, but this approach leverages AFRL resources, and capabilities. Building models and materials databases is beyond the scope of this document, so we will assume this data exists for this discussion.

In order to get to our end goal of generating intensity time-histories at observer points, we need to model how light scatters off the complex 3-D target shape. There are a couple of scale sizes to consider. At the microscopic scale, light reflects off locally flat incremental areas of the target skin according to a Bidirectional Reflectivity Distribution Function (BRDF), which must be measured and exist inside the materials database. At the scale of the target components, the primitive shapes account for local variations of the surface normal, surface areas, local material assignments, and self-shadowing effects. We will resolve all geometry effects at this scale through standard ray tracing techniques.

We need to map how light reflects off the target skin into the full four-pi steradian spherical volume around the target, at every time step of the scenario. We will effectively mesh the entire surface of the target object with small incremental areas, reflect the incident light, and keep a record of how and where this light intercepts a mathematical cross section sphere surrounding the target. The sphere is used for intermediate results bookkeeping and has no physical counterpart. It also has no physical radius, or at least the radius should be considered to be no less than the distance to the nearest observer – which is hopefully in the radiometric far field.

Now, each incremental surface area of the target (some of which is not illuminated) reflects light according to the local BRDF. We will do two obstruction tests to resolve issues with target self-shadowing. The first ray trace is done between the laser source (i.e., the ABL) and the center of each incremental surface patch. We use an aimpoint sampling scheme called *collector points* to represent the center of these small area mesh patches. The first ray-trace test determines whether other parts of the target model (e.g., a fin) shadows (or obstructs) the collector point. If it is obstructed, then no light is assumed to be reflected off that small target area region. We back face cull collector points that are on the opposite side of the target without ray tracing to save execution time. If a collector point is not obstructed, then we map the diffuse scatter onto the cross section sphere, regardless if there might be other obstructing target components in the outbound path of the rays. We do this because diffuse scatter is usually a very minor contribution to the reflected radiometrics, and because this is conservative. We also determine the 2-D coordinates of the collector points as projected into the view port of the laser beam. These coordinates are used to interpolate the incident beam to infer the beam irradiance profile, if the flood beam profile is not used.

A second ray trace is done along the specular reflected ray direction to test for outbound shadowing. If the ray is obstructed, then the light is assumed to become completely diffusely reflected and inconsequential after hitting the obstruction. If the ray is not obstructed then the specular part of the scattered light is mapped onto the cross section sphere, according to the local BRDF. This component of the scatter is the most critical since it is always more intense than the diffuse scatter, mostly because it is more directional.

The collector points are meshed over all surfaces when the 3-D model is built. The spacing should be fairly small to ensure good spatial sampling. We force the area sampling to be done at the primitive surfaces to ensure good sampling, regardless of the view geometry. The other alternative is to just cast many rays at the target. But one gets sampling artifacts going around curved surfaces if this is done. Instead we force uniform sampling at the primitive shape and back project these collector points into the incident beam. While ray tracing is the only viable

technique to use for obstruction testing, one can unfortunately still see small aliasing artifacts. We will use a combination of numerical sampling (ray tracing with adequate spatial sampling) and analytic cross-section scatter equations to minimize the aliasing artifacts one sees.

We ray trace the incident and reflected directions to resolve the target shadowing geometry. We capture these results in what are called scatter-objects. Scatter-objects are the collection of collector points on a primitive by primitive basis. We logically and geometrically group collector points on scatter-objects to cache the scatter from surface patches that are oriented the same.

For flat surfaces (e.g., fins) the specular direction is the same for all points on the surface (i.e., the normal vector does not change). We therefore find out how many collector points are not obstructed (i.e., how much area is exposed) and do a single scatter calculation at the cross section sphere. Figure 1.2 shows this kind of scatter.
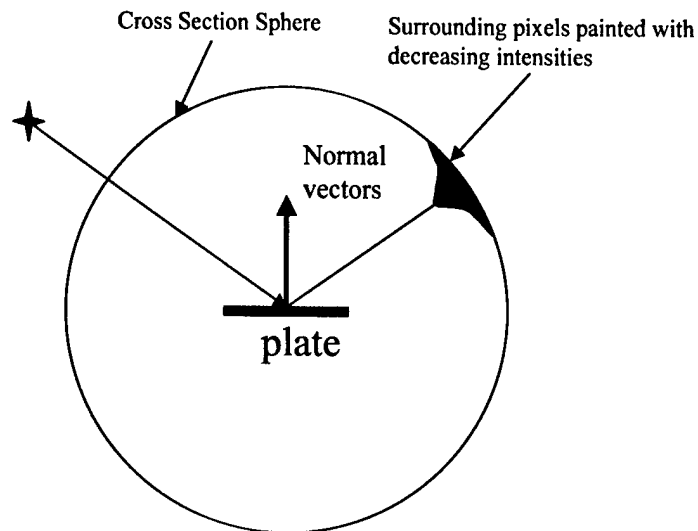
Figure 1.2. Flat Surface Scatter

Singly curved surfaces (e.g., cylinders for the body sections) spread the scatter into a smooth ring around the target. Figure 1.3 shows a simple cut-away of a cylinder and its scatter. Figure 1.4 shows a 3-D view of the scatter ring from a cylinder. The specular BRDF drives the angular spread of the reflected light in this ring. The important point of Figure 1.4 is that the scatter around the ring is smooth. If we use a more traditional approach to mapping the reflected BRDF onto the cross section sphere, where each area element is individually mapped, we can get fairly bad aliasing artifacts if the BRDF is shiny. Discrete sampling around a cylinder leads to discrete normal vector directions, which leads to clumped intensities. By pre-integrating around the cylinder analytically, we avoid this aliasing effect, as was done for Figure 1.4.

Cross Section Sphere

Shiny ring
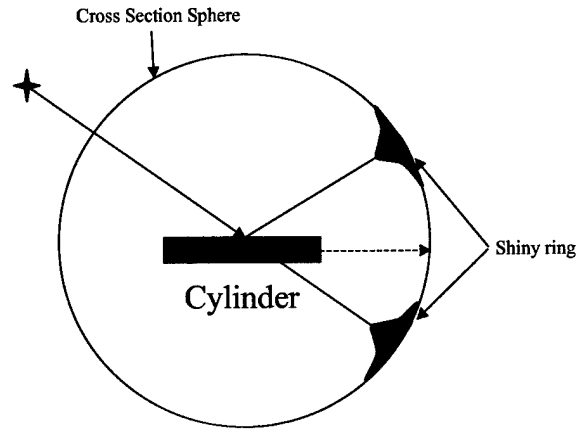
Cylinder

Figure 1.3. Singly Curved Surface Scatter

Cross Section Sphere

Scatter ring
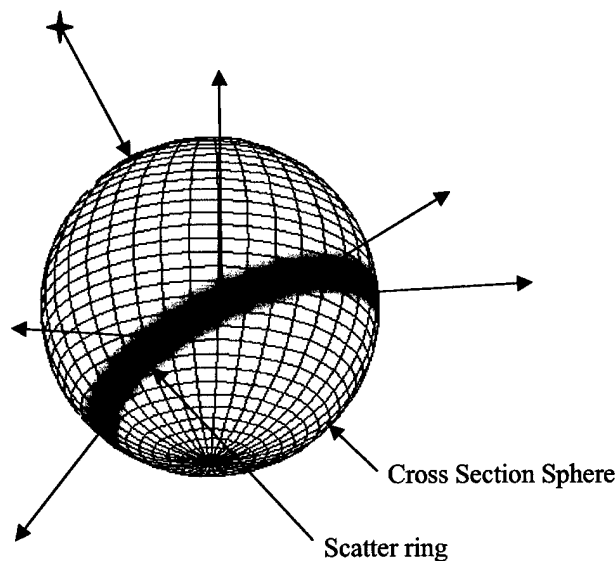
Figure 1.4. 3-D View of Cylinder Scatter

Doubly curved surfaces (e.g., sphere or ellipsoid) scatter light into the full sphere surrounding the shape. The effect is much like diffuse scatter.

Once all cross section spheres have been populated for the full scenario times, we can begin to generate the intensity time-histories for each observer. At each time step we know the position of the ABL lasers, the position of the target missile, the scattered cross sections mapped onto the cross section spheres (by wavelength), and the position and velocity of each observer.

All observers are isolated points in space. When we want to evaluate human eye hazards, we dot the earth with an array of point observers that are conformal with the local terrain, all the way out to the visible horizon, as seen from the target (the scattering center). The requirement is that no hazard may go undetected within an angle of ½ degree, again as seen from the target. Ground observers are therefore spaced apart in angular space by ½ degree, which means they are spaced

further apart near the horizon, and closer together right under the target (i.e., where the range is less). For air corridors and base boundary fences, the meshing is uniform. But, observer points know nothing about other observer points and cause no obstruction. The code is forced to brute force evaluate all observers, one at a time.

Now, for each instance of an observer, to calculate the intensity time-history we freeze time at the first instant a laser comes on. At this time we know all locations (ABL, target, observer), and their orientations. The scattered cross sections at each time step are fixed to the cross section sphere. But between time steps, the scattered distributions change slightly because of the source motion, the target motion, and the observer motion. Figure 1.5 shows a spot at time t_1 (red) and how it has moved to a new location at time t_2 (blue). What we need is to know how the point observer has moved, in one time step, across the surface of the cross section sphere. We then interpolate the cross sections as we traverse this path from time step to time step (i.e., from cross section sphere to sphere). At each point we need to scale the cross section with range changes and laser irradiance to
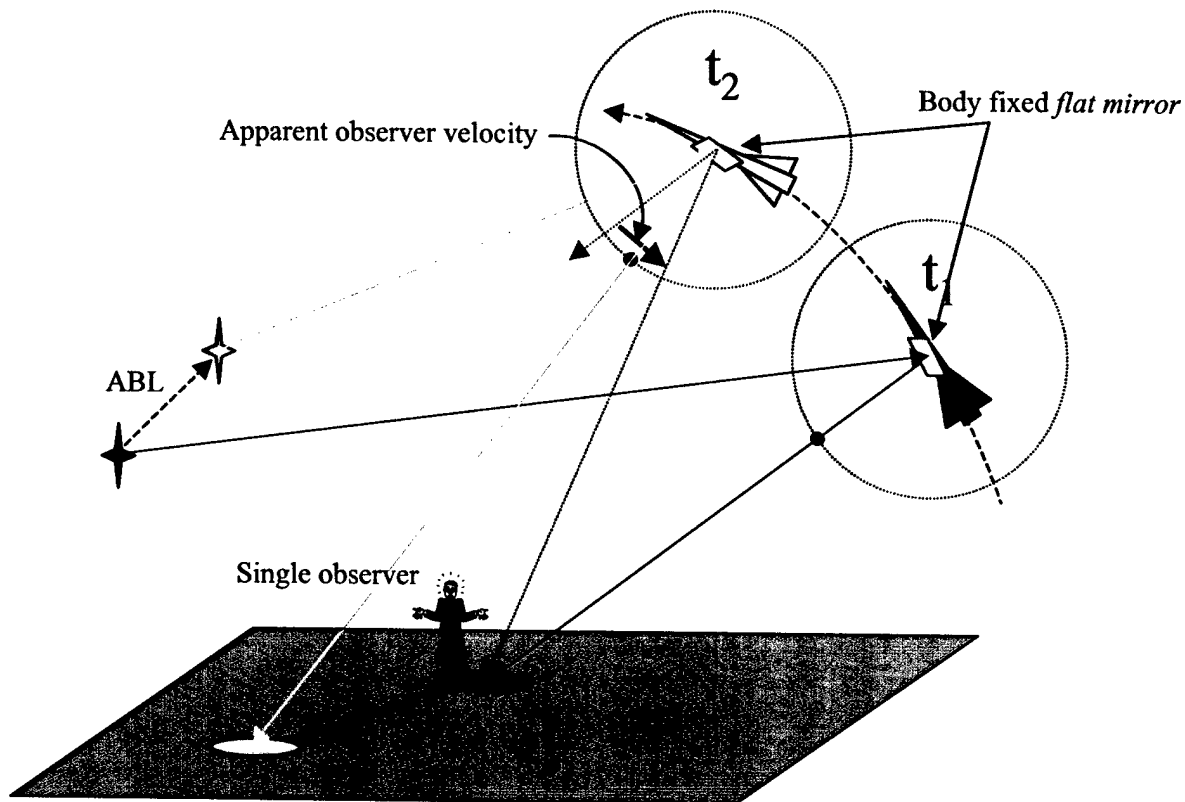
Figure 1.5. Scene Dynamics Showing How a Spot Moves Within One Time Step

get intensities at the observer. Since the angular sampling at the cross section sphere is ½ degree, we need to interpolate (in time) such that the observer position never advances more than one pixel. This way, if the observer passes through a bright single pixel (i.e., a rapid glint) we never miss the event. By the same token, if the cross section sphere is sampled at ½ degree, and the observer moves at an apparent angular rate of say one degree per second, then the smallest minor

time step required would be something on the order of a tenth to a half second – never a millisecond (notice even if the BRDF is shinier than ½ degree, the result is the same). The cross section values interpolated on the cross section sphere are not interpolated in value. When we say interpolate, we are interpolating the viewing direction cosines.

As part of the intensity time-history generated, we need to also scale the intensities with atmospheric extinction. This is a point-to-point transmission loss that must be calculated using an assumed atmospheric extinction profile. If atmospheric transmission effects are not desired, then the transmission is always assumed one.

Finally, the last step is to process the intensity time-history data for each observer against the ANSI standard equations. The intensities are first evaluated to see if any single intensity ever gets above a minimum threshold. If it does not, then an eye hazard can not exist and no further processing of that observer point is necessary, at that wavelength. If the intensities do go above the minimum threshold, then we need to process all the intensity values to see if the Maximum Permissible Exposure (MPE) is truly violated in an integral sense. The time-history is incrementally summed and compared against the ANSI standard to look for hazards. If a hazard should exist, then a hazard object is generated and displayed. Hazards are checked for three exceedance conditions. The first is nominal unaided viewing, where a human is assumed to view the launching missile for a maximum of 10 seconds. The second condition assumes the same person is wearing laser eye protection (LEP). This attenuates the laser intensities, so hazards are less likely to occur. And the third condition assumes the person is viewing the launch through some form of optical aid such as binoculars. Here the potential for a hazard is greater than the first two.

This was a quick summary of the physics algorithms that must be used to complete the hazard calculations in LRST. We will now go through each algorithm in more detail.

2.0 Scenario Geometry Models

2.1 Time Modeling

Time is used and expressed in several important forms throughout the simulation. We first need to adopt coordinated universal time (UTC) as our fundamental basis of modeling time. UTC is the date and time at the Greenwich Prime Meridian. There is no irregular time zone line nor is there daylight savings.

Another important concept is the Julian date, JD, which is the continuous amount of time in days from the epoch January 1, 4713 B.C., noon. The Julian date allows us to cast the usual Gregorian calendar based year-month-day-hour-minute-second time into a single (but long) floating point number. Vallado [1] gives the standard conversions and algorithms to convert between these two. Because it is so difficult to deal with long JD numbers, we always set epochs in the simulation using UTC as year-month-day-hour-minute-second data sets. But internally, many astro-dynamic algorithms require times be expressed as a Julian date.

Internal to any time-driven simulation is the concept of elapsed time, usually in seconds. This is a relative time that is based on some user defined simulation start epoch. In many simulation situations, a relative elapsed time is sufficient. However, when we begin to coordinate simulation predictions with actual real-world dynamical systems, such as missile launches, the sun, the moon, or orbiting satellite trajectories, clearly we need to reference the simulation times to real world UTC times. In general then, we will assume that the simulation time t=0.0 has a precise real world UTC epoch associated with it, like for instance, the missile launch time. The elapsed time we will call time after lift off or TALO.

2.2 Coordinate Frames

All of the calculations in LRST start with predicting the scenario geometry relationships. The ABL scenario fundamentally only needs two kinds of object trajectory prediction, one for the ABL airplane, and one for the missile target. The trajectory represents the motion of the center of mass of an object. In addition to the trajectory position one must know the attitude or orientation angles of the object, that is, its motion about its center of mass. These are the classic six degrees of freedom of a rigid body. We will also want to know the velocity to complete the state vector.

At times we will use one or more of these predicted quantities to evaluate a geometrical relationship in the problem. One common operation we will need is to transform coordinates from one frame to another. This will be done with a 3 by 5 affine transform matrix that we will define later. In order to build this matrix we need both the position and orientation information. In the case of the ABL laser platform, we will also need to define a gimbal coordinate frame to help relate the airplane to the laser beam propagation frame at the nose of the airplane.

As we will show later, only a limited number of trajectory-attitude type combinations make sense. For instance, a missile flies with its cylindrical axis nominally along the velocity vector (with very small attack angles possible).

Before we get into the trajectory physics, we need to establish a useful set of coordinate frames and discuss how the earth should be modeled as a critical part of the scenario. We do not really treat the earth as one of the usual dynamic objects in the simulation, but it nevertheless is a rotating platform upon which all the other objects relate.

There are many useful coordinate frames necessary throughout the simulation. The fundamental coordinate frame is the geocentric equatorial system or earth centered inertial (ECI) frame. This frame is inertial, non-rotating, and so we are assured Newton's equations are satisfied. The x-axis points along the vernal equinox, the z-axis points along the mean earth rotation vector, and the y-axis forms the right-handed triad.

Since the earth rotates approximately around an axis going through the North Pole, we often use a second coordinate frame coincident with the ECI, which rotates about the same z-axis, one revolution per sidereal day. This is called the earth centered - fixed (ECF), - relative (ECR), or - earth-fixed (ECEF) frame. Ground observers, air corridors, base boundaries, the ABL, and the missile trajectories are all calculated relative to this frame. The z-axis is aligned with the ECI frame, but the x and y-axes rotate. This is neither a stationary nor an inertial frame. The ECR x-axis is in the equatorial plane and the Prime Meridian longitude plane, which goes through Greenwich England. The world latitude and longitude lines are fixed to this frame. It turns out to be highly advantageous to work all the LRST geometry in this frame, especially since so many of the observers are spatially fixed. We will also freely convert back and forth between the ECR Cartesian coordinates, and standard latitude/longitude/altitude coordinate parameters within the simulation.

We sometimes will want to generate a local-vertical local-horizontal frame (LVLH), which starts with the z-axis pointing upward, the x-axis points north (except at the north pole where its indeterminate), and the y-axis forms a triad going west. The LVLH frame is generated using just a trajectory coordinate point. As the trajectory changes, the LVLH frame keeps changing, always looking north.

2.3 Earth Modeling

The earth can often be modeled assuming a simple spherical shape with a mean equatorial earth radius (r_e = 6,378.1363 km) [1]. However, we will start by adopting the fundamental shape of the true earth as being an oblate spheroid, which is an ellipsoid of revolution about the *semiminor* or polar z-axis. The derived polar radius is r_p = 6,356.751600 km. The eccentricity of the earth is very small and equal to: ε = 0.081819221456 (or e^2 = 0.00669437999013). We will adopt the shape of the earth at a mean sea level as being:

$$\frac{x^2}{r_e^2} + \frac{y^2}{r_e^2} + \frac{z^2}{r_p^2} = 1 \tag{1}$$

Often times one uses the notation for the three earth axes I,J,K. We will use x,y,z. The earth at the poles is closer to the earth center by about 21 km compared to points on the equator. At a latitude of 35 degrees north (Albuquerque NM, for example), we are closer to the earth center by

about 7 km due to its oblateness. We need to be consistent with this representation of the earth's shape, even in the visualizations.

The actual detailed shape of the earth of course is not exactly ellipsoidal. Local terrain elevations modulate the shape, and strictly speaking the ellipsoid does not always follow the true geoid – a plumb-bob always hangs perpendicular to the geopotential surface geoid. The geoid deviates from the ellipsoid due to mass variations, but never by more than 102 meters. So consequently the ellipsoidal equation above is a very good approximation to the true mean-sea-level earth shape.

A point on the earth's surface is usually specified by three map coordinates: longitude, geodetic latitude (sometimes given as geocentric), and local altitude above a mean sea level (i.e., the earth ellipsoid). Geocentric measures angles from the earth's center. Geodetic latitude is the angle the local surface normal (actually the direction a plumb-bob takes) makes relative to the equatorial plane. The geodetic latitude angle is always slightly larger than the geocentric angle for the same position. Thus the same point has a geocentric angle different from its geodetic latitude angle.

A position P_{site} on the surface in geo<u>centric</u> coordinates would be:

$$\vec{P}_{site} = \begin{vmatrix} r_{site} \cdot \cos(\phi_{gc}) \cdot \cos(\theta) \\ r_{site} \cdot \cos(\phi_{gc}) \cdot \sin(\theta) \\ r_{site} \cdot \sin(\phi_{gc}) \end{vmatrix} \qquad (2.1)$$

where ϕ_{gc} is the geocentric latitude, θ is the longitude, and r_{site} is the radial distance to the ground site point:

$$r_{site} = \frac{r_e \cdot \sqrt{1 - \varepsilon^2}}{\sqrt{1 - \varepsilon^2 \cdot \cos^2(\phi_{gc})}} + h \qquad (2.2)$$

and where h is the local altitude above mean sea level.

If the latitude is given in geo<u>detic</u> units, then finding the position coordinates is a little more difficult. Without producing the entire derivation [1], the Cartesian coordinates for a ground point are <u>defined</u> by:

$$\vec{P}_{site} = \begin{vmatrix} (C_e + h) \cdot \cos(\phi_{gd}) \cdot \cos(\theta) \\ (C_e + h) \cdot \cos(\phi_{gd}) \cdot \sin(\theta) \\ (S_e + h) \cdot \sin(\phi_{gd}) \end{vmatrix} \qquad (2.3)$$

where the two factors C_e and S_e are given by:

$$C_e = \frac{r_e}{\sqrt{1 - \varepsilon^2 \cdot \sin^2(\phi_{gd})}} \tag{2.4}$$

$$S_e = \frac{r_e \cdot (1 - \varepsilon^2)}{\sqrt{1 - \varepsilon^2 \cdot \sin^2(\phi_{gd})}} \tag{2.5}$$

and where ϕ_{gd} is the geodetic latitude. Equation 2.3 is exact.

The inverse problem of inferring the longitude, geodetic latitude, and altitude from a position coordinate is straightforward, but not exact. If we take the ratio of the site position z coordinate to its equatorial coordinate (i.e., the magnitude of the x and y components – recall the ellipsoid is symmetric about a rotation of the minor z-axis, and we can set $w = \sqrt{x^2 + y^2}$):

$$\frac{z}{w} = \frac{\sin(\phi_{gd}) \cdot (1 - \varepsilon^2)}{\cos(\phi_{gd})} \cdot \frac{\dfrac{r_e}{\sqrt{1 - \varepsilon^2 \cdot \sin^2(\phi_{gd})}} + \dfrac{h}{(1 - \varepsilon^2)}}{\dfrac{r_e}{\sqrt{1 - \varepsilon^2 \cdot \sin^2(\phi_{gd})}} + h} \tag{2.6}$$

Since the eccentricity is so small, and because local topographic altitudes are so much less than the earth's radius, we can *approximate* the right hand ratio to 1.0 and solve for the geodetic latitude:

$$\phi_{gd} = \arctan\left(\frac{z}{w \cdot (1 - \varepsilon^2)} \right) = \arctan\left(\frac{z}{\sqrt{x^2 + y^2} \cdot (1 - \varepsilon^2)} \right) \tag{2.7}$$

The ratio of $\dfrac{z}{w} = \tan(\phi_{gc})$ is the tangent of the geocentric latitude. We conclude that the conversion from geocentric latitude to geodetic latitude is approximately:

$$\tan(\phi_{gc}) = \tan(\phi_{gd}) \cdot (1 - \varepsilon^2) \tag{2.8}$$

One can convert back and forth between the two angles easily using Equation 2.8. The error in Equation 2.7 is zero for sea level points (where h=0), and increases slightly with altitude. Latitudes are always zero at the equatorial plane, and go positive to 90 degrees at the north pole, and negative toward the south. How accurate is Equation 2.7?
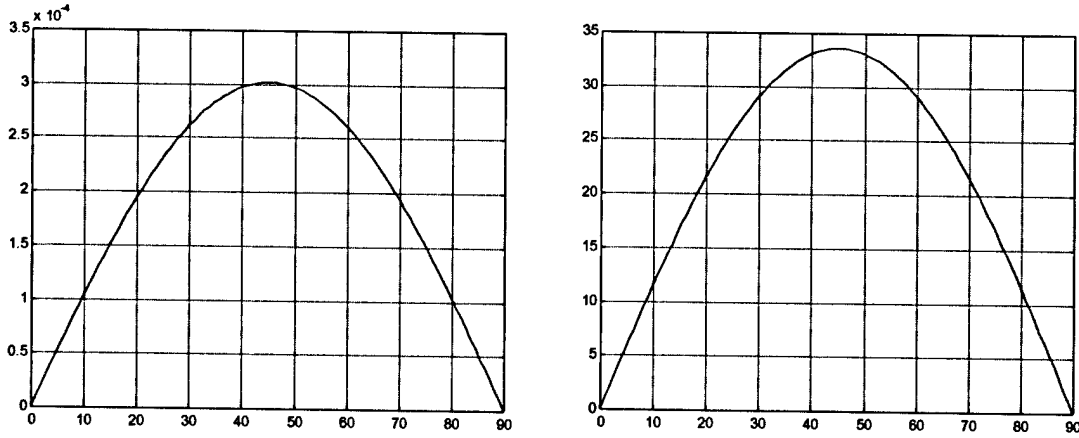
Figure 2.1. Error Plots When Longitude-Latitude-Altitude Coordinates are Converted to (x,y,z) and Then Back as a Function of Geodetic Latitude When the Local Altitude is 10 km

We start with parametric geodetic latitude angles, transform to (x,y,z) coordinates (exact) and then back to longitude-latitude-altitude (approximated by Equation 2.7). We default the local altitude to 10 km. The full loop angles are compared with the starting angles and plotted on the left in Figure 2.1. The agreement is excellent. On the right in Figure 2.1 is the error in the distance of the point on the earth surface going full loop. Both errors seem to be linear in the local altitude (zero altitude produces zero error!). For low altitudes the error is tens of meters. To get accuracies better than this we must use an iterative algorithm outlined in [1]. The error then is under a millimeter for all altitudes, even out to space.

The altitude is the difference between the site position radius and the mean sea level radius at the respective geodetic latitude: $alt = r_{po\ int} - r_{MSL}$ where we have:

$$r_{po\ int} = \sqrt{x^2 + y^2 + z^2} \tag{2.9}$$

and,

$$r_{MSL} = r_e \cdot \sqrt{\frac{\cos^2(\phi_{gd}) + (1 - \varepsilon^2)^2 \cdot \sin^2(\phi_{gd})}{1 - \varepsilon^2 \cdot \sin^2(\phi_{gd})}} \tag{2.10}$$

The error in the altitude solution when going full loop at 10 km altitude is less than 5cm, so the point separation error is entirely found in the latitude error solution (i.e., along the earth surface) and therefore not in the radial direction.

The longitude angle is given simply by:

$$\theta = \arctan(\frac{y}{x}) \tag{2.11}$$

except at the poles where longitude is undefined. Longitudes are defined as starting at zero at the Greenwich Prime Meridian, and positive angles go eastward with a right handed earth coordinate system. Westward going longitudes are usually designated with negative angles.

A static point in ECR coordinates spins with the earth motion. To determine the transform from ECR to ECI, we need to know the angle the Greenwich Meridian has rotated (goes eastward) relative to the ECI x-axis. We convert the Julian date to centuries since midnight January 1, 2000:

$$Tu = (JD_{int\ egral\ -days} - 2451544.5)/36525 \qquad (2.12)$$

the angle the earth has rotated to the nearest whole day is:

$$\theta_c = 1.753368559 + 628.3319706889 \cdot Tu +$$
$$6.770708127\ e-6 \cdot Tu^2 - \qquad (2.13)$$
$$4.5087672343\ 1868\ e-10 \cdot Tu^3$$

Finally, add the rotation angle the earth has undergone during the remaining fractional day:

$$GST = \theta_c + \omega_e \cdot (JD_{fraction} - 0.5) \qquad (2.14)$$

where the current earth's rotation rate is 6.30038809866574 radians/solar day. GST is the Greenwich Sidereal Time, and the units are radians.

The GST is next added to the fixed local longitude. This angle then allows us to form a transform between local point coordinates and world ECI coordinates.

2.4 Terrain Modeling

Terrain altitude variations are actually important to the LRST modeling for several reasons. Depending where on the world the scenario is centered, the local terrain elevations can become quite high (White Sands Missile Range, for example). This places observers potentially closer to the target missile, and at elevations where the atmospheric attenuation is less. For low flying missiles (for instance, 1-2 km maximum altitude above local terrain), one might want to consider terrain obstruction or masking to be important.

We use a whole-world DTED (digital terrain elevation database) database for elevation modeling. Elevations are sampled on half arc-minute separations, which is about 927 m (0.62 mile) resolution. The database is derived from higher resolution DTED sources, so there will be some aliasing and/or averaging of the terrain.

We looked at the issue of terrain masking to see how important this would be to typical ABL scenarios. The typical TBM altitude is something above at least 10 km (where cloud cover begins to end). Small tactical rockets (Katyusha) might only reach 1 km in altitude. The question is, how much terrain masking might we expect for these two cases?

We chose a point somewhere near White Sands National Park (-106.4 degrees West, 32.75 degrees North) to look at how much the local mountains might obstruct reflected rays scattering from a missile target. Figure 2.2 shows a 120-km wide graphic of the White Sands region with greatly exaggerated mountains (7x actual).
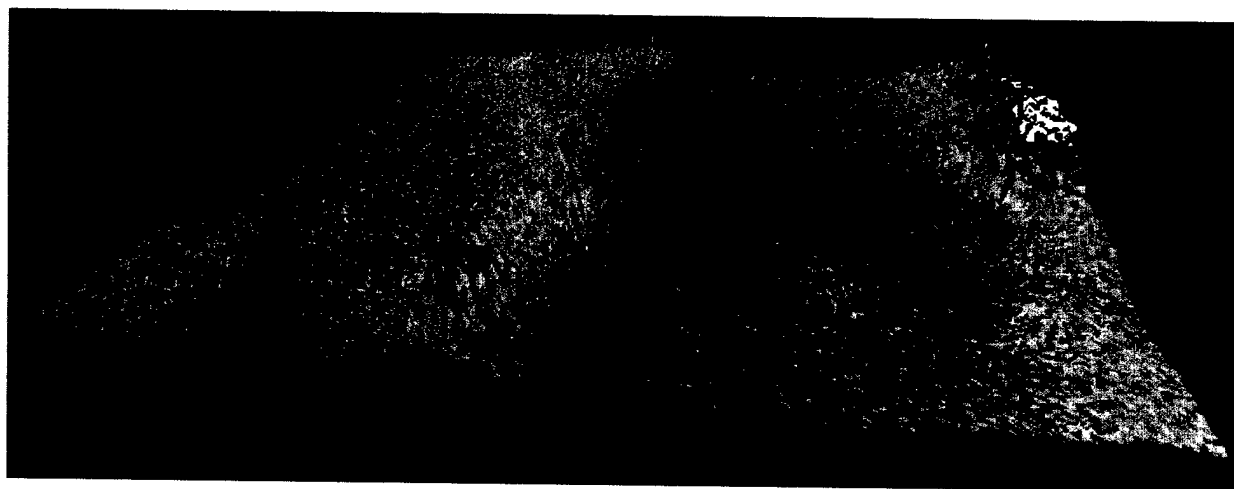
Figure 2.2. White Sands Terrain Masking Example

Figure 2.3 shows the terrain polygons floating over an array of point observers. The yellow dot is a target point located 1 km above the local terrain (left), and 15 km (right). The green and red dots are observers located just a few meters above the actual local terrain (as defined by the terrain database). Triangular polygons connect vertex points where the altitudes are sampled. The resolution of the terrain is 927 meters between vertices (observers). The Figure 2.2 terrain polygons are now being seen edge on.

Rays were traced from the yellow dot (1 km and 15 km high) to every observer point. Green dots represent observers with an unobstructed LOS to the target dot. Red dots represent observers where terrain has obstructed the view of the missile point. Shadow regions behind the mountains are easily seen as red dots (left panel). 40% of the observers are obstructed (red) when the missile is 1 km high (left panel). But only 0.15% of the observers were obstructed when the missile was 15 km high (right panel).

We obviously picked a case with local mountains on either side of a valley to get maximum obstruction at low altitudes. When the terrain is less mountainous, even low altitude targets will experience minimal terrain obstruction.
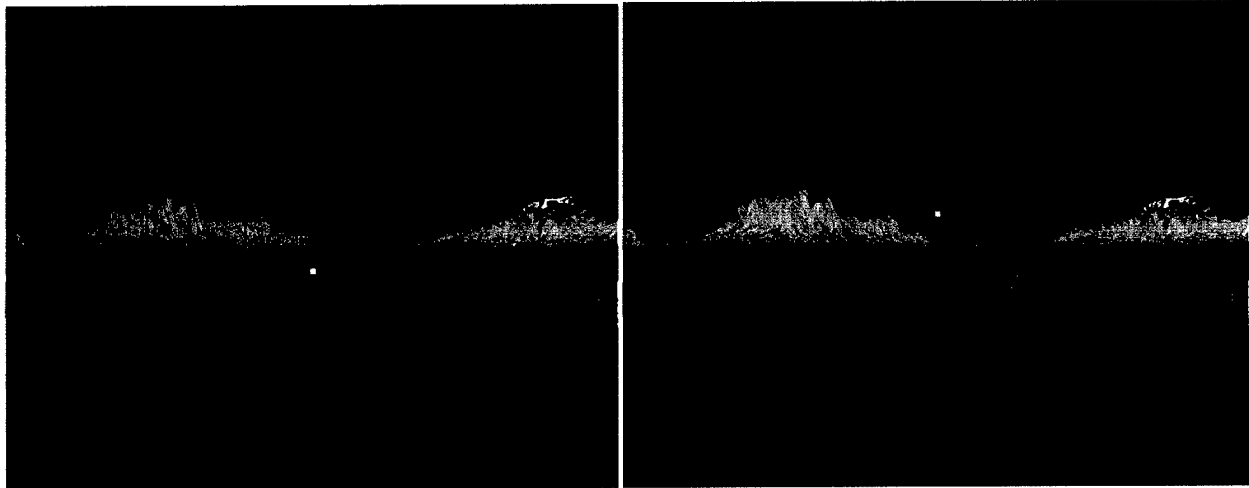
Figure 2.3. Left is a 1 km Altitude Target, Right is a 15 km Altitude Target

In Figure 2.4 we show an end view of the same terrain, but with true elevations in correct perspective, with the target at 15 km altitude. There should be little wonder why virtually nothing is masked. Near the horizon (which these examples do not reach) one might expect slightly more terrain obstruction due mostly to natural earth curvature effects. In general we are less interested in these points.
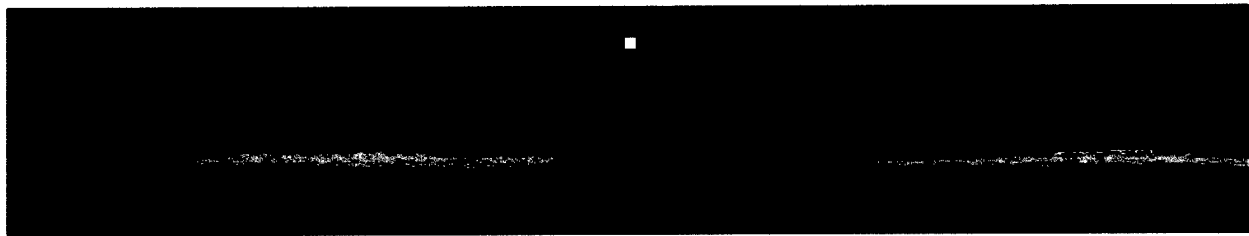
Figure 2.4. End View With Terrain Displayed With True Altitudes

When we varied the missile altitudes and checked for terrain obstruction, we found that less than 5% of observer points were masked for target altitudes above 4 km. The important conclusion we take away from this analysis is that at typical ABL scenario altitudes (i.e., 10 km and above), very few ground observers are actually shadowed by local terrain. Therefore, in general we should not include terrain masking in LRST for this reason. Besides, not allowing masking is a more conservative approach to our modeling. Only very low missile targets in mountainous regions manifest significant terrain obstruction, which are going to be very trajectory sensitive. In other words we can not count of terrain conditions to safely obstruct the observer line of sight (LOS) to the target.

2.5 Target Missile Trajectory Modeling

At the missile we define a body coordinate system that is fixed to the physical mechanical model. The missile CAD models must have a coordinate convention where the model x-axis is meant to

point along the velocity direction. The relationship, i.e. the rotational transform, between this body frame and the world ECR frame is the attitude of the missile. LRST does not predict missile fly-out profiles on-line. The code imports the trajectory and attitude data from an external file. Figure 2.5 shows a file format example for a missile flight path. The code first reads the entire table of values. Then as the simulation progresses, values are linearly interpolated between discrete times.

```
ECI Coordinate System Output Release Sep0898/TBK
 time (sec)    Alt(m)      Long (deg)   GDlat (deg)   VEast(m/s)    VN(m/s)      VD(m/s)     Heading(deg)  Elevation          Roll
    0.00     1.000E+00  -0.120E+03   0.348E+02   0.000E+00   0.000E+00   0.000E+00  -0.899E+02   0.899E+02   0.520E-02
    1.00     0.326E+02  -0.120E+03   0.348E+02   0.000E+00   0.000E+00  -0.348E+01  -0.898E+02   0.899E+02   0.145E+00
    2.00     0.422E+02  -0.120E+03   0.348E+02   0.000E+00  -0.100E-01  -0.159E+02  -0.674E+02   0.899E+02   0.225E+02
    3.00     0.641E+02  -0.120E+03   0.348E+02   0.000E+00  -0.100E-01  -0.286E+02  -0.328E+02   0.899E+02   0.571E+02
```

Figure 2.5. Raw Missile Trajectory Data Format Example

The latitudes are assumed in geodetic coordinates (note the GDlat in the table). The velocity components are not actually needed, and incidentally were meaningless in the data from this table. The attitude is constructed from the last three columns of angles. The heading is the azimuth angle between the flight plane and north (the convention uses a left handed rotation going east from north). Thus the –89 degrees sends this missile out *westward* from Vandenberg AFB. The elevation column determines the elevation angle up from the local horizontal (LVLH) plane for the missile x-axis. Thus the heading/elevation angles determine the flight direction cosines. The last roll column determines how the missile is rolled about the flight direction. Figure 2.6 shows the missile body frame 60 seconds after lift-off. The solid yellow line is the trajectory path (the red line is the laser beam LOS).

The launch point can be easily translated to a new point simply by adding a longitude and/or a latitude offset. The fly-out azimuth can be changed by introducing a rotation about the local zenith.
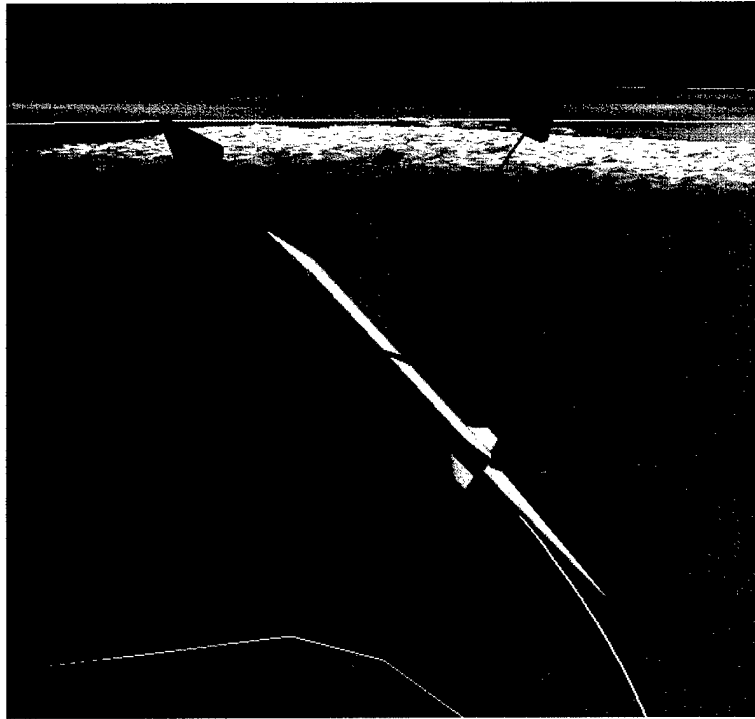
Figure 2.6. Target Body Axes Definition, Where TALO = 60 Seconds

The trajectory can only be followed up to the point where there is no more data. We do not attempt to model a ballistic coast beyond the last point of data. We also do a linear interpolation of the latitude/longitude/altitude data, not a spline. Higher order mechanics like vibrations and flexures of the missile in flight are not modeled, and are assumed to be smaller than the ½ degree resolution we model for the radiometrics.

The target position is the apparent source point for all scattered light. The higher the missile is when being illuminated, the more of the earth surface that can be exposed. As seen from the center of the earth, the angle out to the horizon when the missile is at an altitude of h is given by:

$$\theta \approx \cos^{-1}(\frac{R_e}{R_e + h})$$

(2.14)

At 60 seconds after lift-off, when the missile is about 27 km high, the visible horizon is actually quite distant. Figure 2.7 shows an overhead view of the earth at this point. The latitude/longitude lines are spaced two degrees apart. The horizon circle covers more than 10 degrees in latitude. The terrain polygon model only covers five degrees in this view. So, how much of the earth should we normally display? Five degrees in width is probably enough since intensity falls off as the distance squared. We hardly expect hazards to occur in Nevada and not somewhere close within our limited displayed world view. As far as the ground observer coverage is concerned, however, we must instrument all the way out to the horizon.
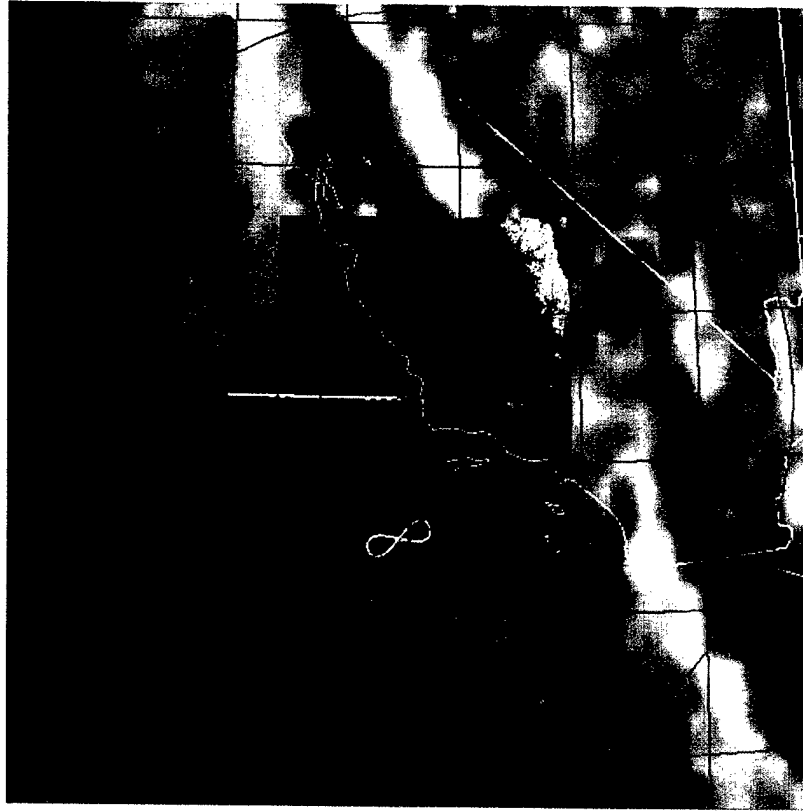
Figure 2.7. 60 Seconds After Lift-off, 27 km High

2.6 Airborne Laser Airplane Trajectory Modeling

We define a body coordinate system for the ABL that is fixed to the physical mechanical model. An airplane trajectory is defined by an initial location, a speed and a heading. The initial location is defined by a geodetic latitude, longitude, and an altitude. The heading is the angle the velocity vector makes in the LVLH frame eastward from north (again, a left-handed rotation starting north and going east). Airplanes fly an earth fixed, great circle. A great circle is an approximate circle in a plane that contains the center of the earth. The flight path is relative to a fixed ECR coordinate frame. We say approximate because we constantly adjust the local altitude to be constant. Thus the true trajectory around the earth would be an ellipsoid. The ABL will probably never fly directly at a target missile.

Between the ABL and the missile is a gimballed telescope that accurately points the laser beam LOS at the missile. We therefore need to add a telescope coordinate frame between the ABL and the missile. Telescope optics are usually mechanically pointed using gimbal systems. The gimbal system we consider here is a two-axis geometry, with one inner rotation axis and one outer axis at right angles to the first. The outer azimuth axis is aligned with the nominal ABL velocity direction, and the inner axis points the telescope fore and aft. Since the laser propagation model assumes circular symmetry, we only need to know the exact pointing LOS from ABL to missile.

3.0 Target 3-D Modeling

The target model must be constructed mathematically in three dimensions. We use the AFRL Satellite Assessment Center's (SatAC) solids paradigm for modeling 3-D objects. A computer aided design (CAD) tool called Solids Modeling Tool (SMT) is used to define solid primitives and position/rotate them in the model body coordinates. The modeling interface defines eleven solid primitive types: box, right cone, right cylinder, elliptical cone, elliptical cylinder, ellipsoid, hemisphere, hexahedron, rectangular parallel piped, sphere, and wedge. An example of each is shown in Figure 3.1, starting in the upper left and proceeding down to the lower right.
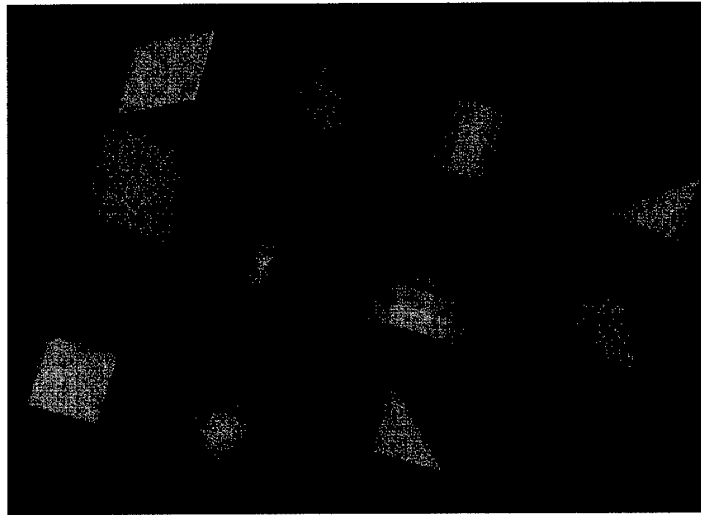
Figure 3.1. Suite of 11 Solid Primitives Used In SMT

Each primitive is a convex closed solid, with from one to six faces. Most missile models can be built with some combination these shapes.

Occasionally we need to model minor complexity by combining these shapes together using Boolean operators. This is called constructive solid geometry (CSG). The three algebraic operators are Union (which adds solids together), Subtraction (one solid removes material from another solid), and Intersection (only the common overlap volume of two or more solids is seen). CSG algebra is beyond the scope of this discussion.

Figure 3.2 shows an example of CSG subtraction. The final solid starts with a solid cube. Then a cylinder is subtracted through the middle. Finally a sphere is used to remove one corner. The rendered solid is shown on the left, and the wire frame is on the right. The dashed lines represent subtracting solids (voids).
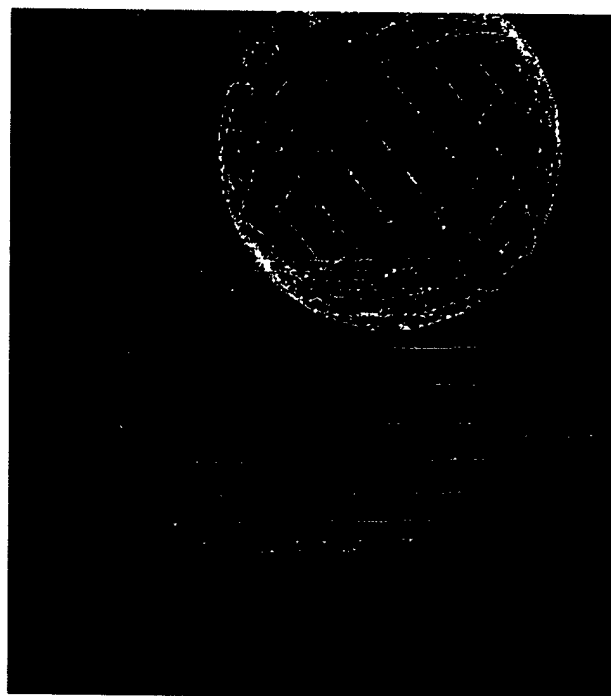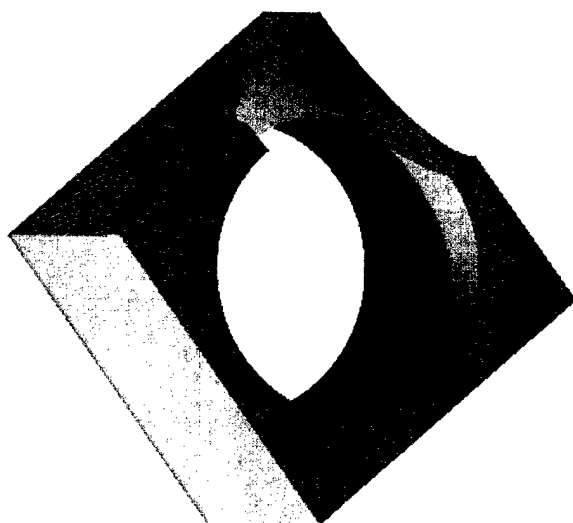
Figure 3.2. Example of the CSG Subtraction Operator

Simple solid shapes and CSG work well for man-made objects, especially simple shapes like missiles. Very complex shapes can be built since the modeling language supports hierarchical trees of CSG solids. Complex target models might contain many thousands of primitive solids.

The primitive solids, CSG operators, and coordinate transforms define the physical geometry of the target object. In addition the modeling language supports the modeler assigning material properties to exterior surfaces. The material assignments are done by material code that references back to a materials database called MATTER.DAT. Each *Model Element* (a collection of primitives combined using CSG) is assigned a surface material code, and possibly a BRDF material code. If the BRDF code is missing, it is always assumed to equal the surface code. Also a bulk thermal-mechanical material code is assigned. Each Model Element is also assigned an initial temperature. Local material thickness is derived by doing many ray trace operations perpendicular to a surface. The difference between the entrance and exit intercepts determines the thickness.

Multiple aimpoints or body-points can be assigned using SMT. These are given in both local and global body coordinates. We will use these body coordinates to direct a high power laser beam during execution of LRST. Since we will not be modeling the complex process of tracking and pointing of the HEL beam, we will direct the beam exactly at one aimpoint.

As part of the ray tracing done in LRST, we need to mesh the primitive surfaces with points to properly sample surface areas. We call these mesh points *collector points*. The algorithm for meshing surfaces depends on the primitive shape. We use a texture transform to convert from parametric coordinates to primitive surface 3-D coordinates as part of this meshing process. When primitives are combined using CSG operators, some collector points might be removed by

a CSG operation. To remove these now non-existent collector points we do an initialization ray trace to see if there is any material within 1 mm of each collected point (the points are always moved 1 mm away from material surfaces to avoid ray trace precision problems). When an entrance point is more than 1 mm, the collector point is assumed nonexistent and removed from the model.

Figure 3.3 shows an example of a cylinder meshed with collector points. The collect point spacing should, in general, be reasonably close to ensure capturing local geometric details. Cones and cylinders have a minimum of 200 collector points meshed around their periphery, regardless of the mesh spacing along the axis of revolution direction.
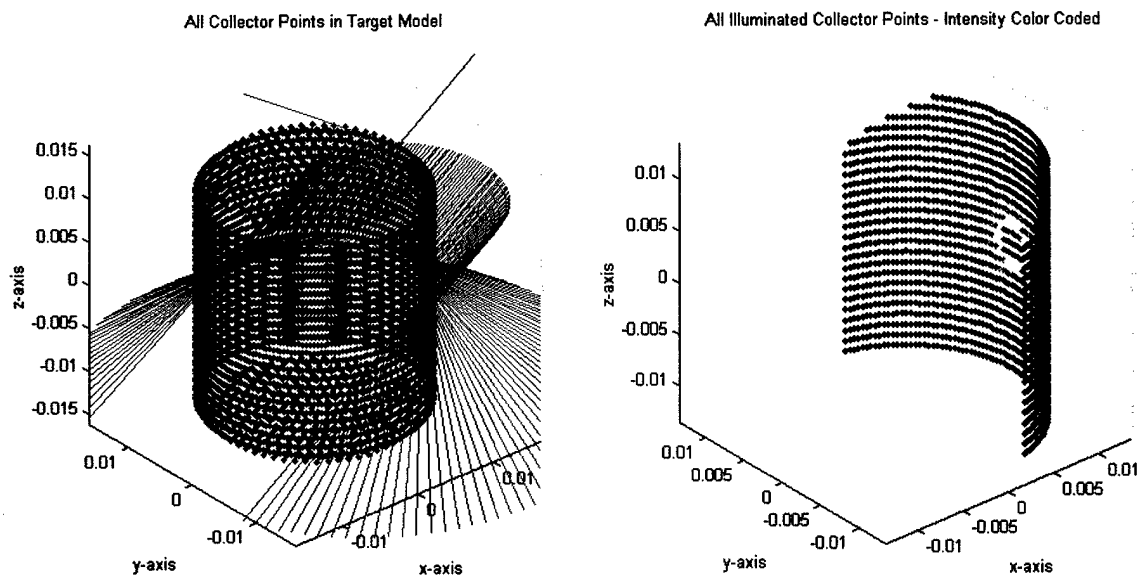
Figure 3.3. Cylinder meshed with collector points points (left: blue, right: multi-colored)

Incident rays (red) and reflected rays (green) are shown on the left. On the right, the collector points exposed to the incident laser beam are colored according to the local beam irradiance. Collector points not irradiated are now shown.

Figure 3.4 shows a missile model meshed with collector points, spaced rather far apart to demonstrate their location. Rays are only traced to collector points. The radiometrics are based on which collector points are obstructed from either the incident beam or the specular reflection direction. The closer collector points are spaced, the more it takes to cover a model (which can be in the tens of thousands), the more rays which must be traced to do the radiometrics, and the more radiometric calculations that must be done. If structural detail exists in the target model, then more points will resolve the potential obstructions. So, there is a trade-off between spatial resolution, and fidelity, and of course computational cost.

The rays that are traced to the collector points are basically the same ones visualized by LRST, except we thin the number of displayed rays to minimize the display confusion. For example, in Figure 3.4, only one row of rays is shown for the entire length of the missile body cylinder, or its nose cone. The fins only have one reflected ray per face.
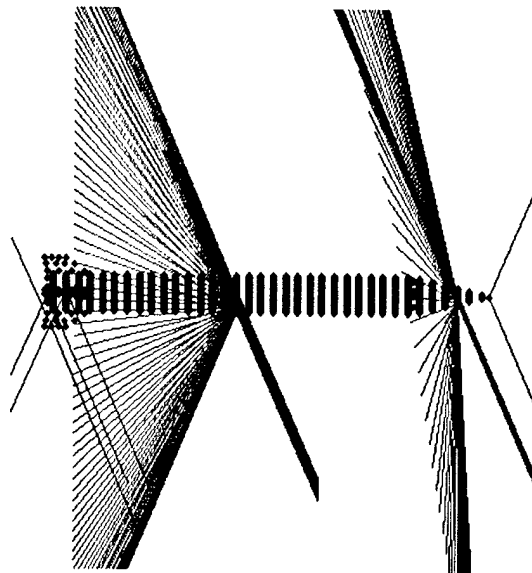
Figure 3.4. Full Missile Model With Collector Points Spaced Far Apart

While the modeling format supports texturing and subassembly articulation, these are neither implemented, nor needed by LRST.

Figure 3.5 shows a typical missile model built using SMT. The primitives and subassemblies define the 3-D shape of the target object. The model in Figure 3.5 had gold plating assigned on its exterior surface.

Figure 3.5. Missile Model Built Using SMT

If a high power laser is used to heat a target model (e.g., if the ABL were to lase a missile during a test scenario), then over time the exterior surface material properties are very likely to change

with temperature. If the test were successful, the missile would melt and break up during flight. These physical changes to the missile model are not modeled in LRST. We will assume the target description remains constant during the test flight.

For low power illuminator lasers, this is clearly the case. For high power situations, this is not the case. Target break-up will probably pull the target out of the HEL beam within a few milliseconds due to aerodynamic drag. But, during the dwell time when the exterior surface is being heated, the changing material properties will scatter light differently. We can either assume surfaces will become darker (less reflective) and more diffuse, or we can assign the worst case *hot* BRDF material properties to ensure the radiometrics are conservative.

3.1 Ray Tracing

Ray tracing starts with the definition of a ray. Rays are essentially infinite lines in 3-D space with a directionality. Thus, a ray is defined by a point in 3-D space $\vec{x}(x,y,z)$, and the pointing direction cosines $\hat{d}(l,m,n)$. We can propagate the ray forward by a distance A to a new point $\vec{x}_1(x,y,z)$ by:

$$\vec{x}_1 = \vec{x}_o + A \cdot \hat{d} = (x_o + A \cdot l)\hat{i} + (y_o + A \cdot m)\hat{j} + (z_o + A \cdot n)\hat{k} \tag{3.1}$$

When we refer to ray tracing, we really are talking about *intercepting* a ray with a solid primitive object. For the kinds of closed convex solid primitives used in our paradigm, there are 0, 1 or a maximum of 2 intercepts possible with any one primitive solid. If the ray line does not intersect the solid, then there are of course zero intercepts. If the ray-line just touches the solid at a tangent point, then there is just 1 (degenerate) intercept. We are really free to classify this kind of intercept as not really being an intercept (it would be very difficult numerically for this to happen anyway) at all. If the ray passes through the solid then there will be one entrance and one exit intercept. Figure 3.6 shows a simple ray-sphere intersection example. The ray starts at the upper left and travels along the direction of the arrow (i.e., to the lower right). The distance to the first entrance intercept is A_1, and the distance to the second intercept is A_2. The respective local surface normal at each intercept is also drawn.
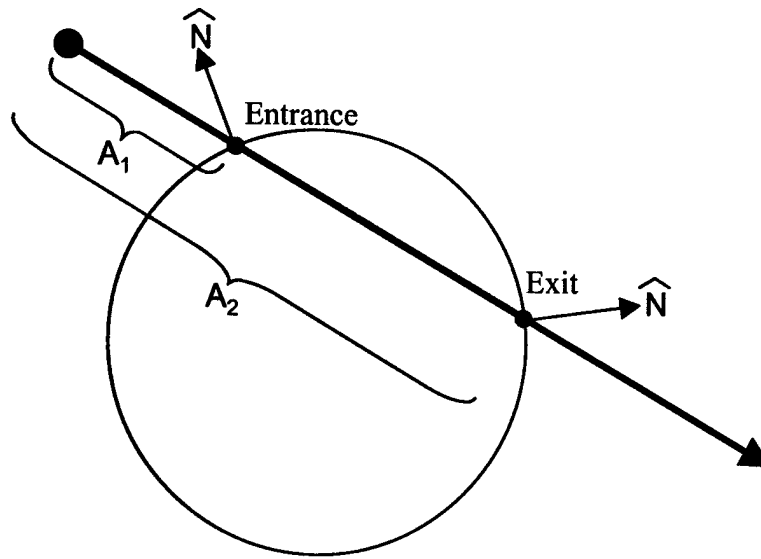
Figure 3.6. Ray-Sphere Intersection Showing Two Intercept Points

Tracing rays mostly involves finding the distances A_1 and A_2 from the ray starting point to each intercept. For quadric primitives this involves solving a quadratic equation, for planes it's a simple linear equation. We solve for the distance to an intercept by starting with the equations for a line and the primitive surface. The parametric equation for a line in three dimensions is previously given in Equation 3.1. The equation for a primitive surface depends on the type of surface.

In addition to the distances to intercepts, we will also want the unit vector of the local surface normal at the intercept. This normal vector is also a function of the primitive surface shape.

Finally, we sometimes need to know the parametric or texture coordinates for an intercept point. In the sections below we will present intercept equations, surface normal equations, and texture coordinate equations for each of the primitive shapes. Some of the solids are formed using planes, so we will start with a plane.

3.2 Plane

The equation of a plane is:

$$a \cdot x + b \cdot y + c \cdot z + d = 0 \qquad (3.2)$$

where (a,b,c,d) are constants that define the plane, and (x,y,z) are the independent variables for any point in the plane. Usually, one casts (a,b,c) as the direction cosines of the normal vector to the plane.
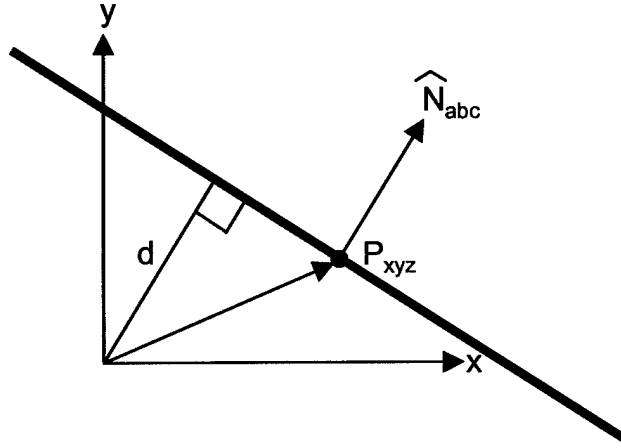
Figure 3.7. Two-dimensional Plane

In Figure 3.7, we see that the perpendicular distance d from the coordinate origin to the plane is given by the dot product of any point in the plane $\vec{P}(x, y, z)$, and the unit surface normal vector $\hat{N}(a, b, c)$:

$$\left| \hat{N} \circ \vec{P} \right| = \left| a \cdot x + b \cdot y + c \cdot z \right| = \left| d \right| \qquad \text{(m)} \qquad (3.3)$$

Notice that if the surface normal were reversed, the sign of d would change. Multiplying Equation 3.2 by a minus one does not alter the plane. But, in general, we do want to pay close attention to how we chose the direction of the plane normal. When a plane defines one of the surfaces of a solid, for instance, then the surface normal shall always point outwards.

Now, using Equation 1 for a ray, we can propagate the ray forward to an intercept with the plane by a distance A such that:

$$(\vec{x}_o + A \cdot \hat{d}) \circ \hat{N} + d = (x_o a + y_o b + z_o c + d) + A \cdot (la + mb + nc) = 0 \qquad (3.4)$$

Rearranging for the intercept distance A:

$$A = \frac{-(x_o a + y_o b + z_o c + d)}{(la + mb + nc)} \qquad \text{(m)} \qquad (3.5)$$

Knowing intercept distance A and using Equation 3.1, we can find the intercept coordinate point. If the ray is parallel to the plane, then the denominator in Equation 3.5 is (near) zero. If the plane is *behind* the ray starting point, then intercept distance A becomes a negative number.

We define a transform from physical primitive surface 3-D coordinates to a 2-D parametric texture coordinate space for each primitive type. We refer to this as a texturing transform. The transform is usually set up to allow one to navigate across a 3-D surface by specifying 2-D parametric coordinates (s,t) that vary from 0 to 1 along each axis, and vice versa. Figure 3.8

shows the basic concept. A common example would be for applying a Mercator projection map onto a spherical earth. The map is represented by points in a 2-D space, which uniquely have a 3-D position on the sphere. The most common usage for this transform is to do texture mapping in computer graphics. Here, we want a simple mapping between a 3-D and a 2-D space that completely covers the primitive surface fairly uniformly. We will allocate collector points uniformly in rectilinear texture space and trust that they are evenly spaced in 3-D primitive space. Clearly this will not be the case for such primitive shapes as spheres and ellipsoids. At worst, we will over sample these primitives at their poles. The consequence of over-sampling is computational burden, but does not really introduce other undesirable simulation artifacts.
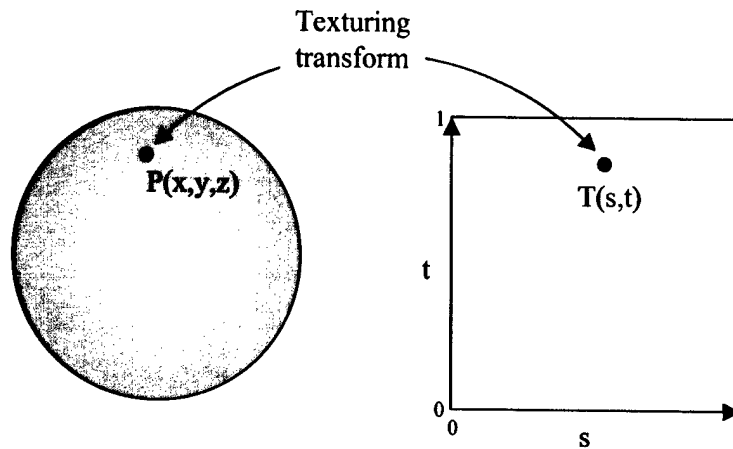
Figure 3.8. 3-D Physical Space Transform to 2-D Texture Space and Back

The transform from physical primitive space to parametric texture coordinate space for a plane depends on the type of plane. We use four different methods of bounding a plane: a rectangle, a quadrilateral, a circular disk, and an ellipse.

For a rectangle, the transform merely stretches linearly the parametric space to completely cover the rectangle along each respective axis. The transform from primitive to texture space is:

$$s = \frac{(x - x_{min})}{(x_{max} - x_{min})}$$
$$t = \frac{(y - y_{min})}{(y_{max} - y_{min})}$$

(3.6)

where x_{min} is the minimum edge coordinate along the first (s) axis, x_{max} is the maximum edge coordinate, y_{min} is the minimum edge coordinate along the second (t) axis, y_{max} is the maximum edge coordinate. For all planes we need to define a dominate plane in order to assign which of the three coordinate axis goes along s and t. We do this by finding which of the principal planes (x, y, or z) has the greatest surface normal projection. For example, if the plane is nearly in the x-y plane, then the s-direction is along x, and the t-direction is along y.

To transform from texture space back to 3-D space we invert Equation 3.6:

$$x = s \cdot (x_{max} - x_{min}) + x_{min}$$
$$y = t \cdot (y_{max} - y_{min}) + y_{min}$$

(3.7)

where again the x and y variables correspond to the respective axis based on the dominate plane.

For a quadrilateral we use a bi-linear interpolation (stretching) algorithm. A quadrilateral is defined by its four vertex points, V_1-V_4. We start by defining eight texture parameters:

$$p_1 = V_{1x}$$
$$p_2 = V_{1y}$$
$$p_3 = V_{2x} - V_{1x}$$
$$p_4 = V_{4x} - V_{1x}$$
$$p_5 = V_{1x} - V_{2x} + V_{3x} - V_{4x}$$
$$p_6 = V_{2y} - V_{1y}$$
$$p_7 = V_{4y} - V_{1y}$$
$$p_8 = V_{1y} - V_{2y} + V_{3y} - V_{4y}$$

(3.8)

The transform from primitive to texture space starts by pre-calculating some values:

$$AA = p_5 \cdot p_7 - p_4 \cdot p_8$$
$$BB = (p_2 - y) \cdot p_5 - p_4 \cdot p_6 + p_7 \cdot p_3 + (x - p_1) \cdot p_8$$
$$CC = (p_2 - y) \cdot p_3 + (x - p_1) \cdot p_6$$

(3.9)

Form the quadratic radical, takes its square root:

$$rad = \sqrt{BB^2 - 4 \cdot AA \cdot CC}$$

(3.10)

Use the negative root if BB is less than zero (i.e., use –rad). Solve for t first:

$$t = \frac{CC}{-0.5 \cdot (BB + rad)}$$

(3.11)

We know $0 \le t \le 1$, so if the denominator is too small, then linearize the quadratic to get the other solution:

$$t = \frac{CC}{AA}$$

(3.12)

If this denominator is also too small, then set t to zero. The first coordinate s is derived from t:

$$s = \frac{(x - p_1 - p_4 \cdot t)}{p_3 + p_5 \cdot t} \tag{3.13}$$

To transform from texture space back to 3-D space:

$$x = p_1 + p_3 \cdot s + p_4 \cdot t + p_5 \cdot s \cdot t$$
$$y = p_2 + p_6 \cdot s + p_7 \cdot t + p_8 \cdot s \cdot t \tag{3.14}$$

This gives us the projected two coordinates in the dominate plane. We then need to solve for the third 3-D coordinate using the equation of the plane:

$$z = \frac{-(a \cdot x + b \cdot y + d)}{c} \tag{3.15}$$

where these equations assumed the dominant plane was the x-y plane.

For a circular disk, we use a simple rectilinear to polar transform (i.e., concentric circles). The s-direction is around a circle, and t is along the radius. The transform from primitive to texture space starts by calculating the radius of the (x,y) projected point, and its azimuth:

$$r = \sqrt{x^2 + y^2}$$
$$\theta = a \tan 2(y, x) \tag{3.16}$$

Then the texture coordinates are:

$$s = \frac{(\theta + \pi)}{2 \cdot \pi}$$
$$t = \frac{r}{R} \tag{3.17}$$

where R is the full radius of the disk.

To transform from texture space back to 3-D space:

$$\theta = s \cdot 2 \cdot \pi - \pi$$
$$r = t \cdot R$$
$$x = r \cdot \cos(\theta)$$
$$y = r \cdot \sin(\theta) \tag{3.18}$$

and the last coordinate is:

$$z = \frac{-d}{c} \tag{3.19}$$

where these equations again assumed the dominant plane was the x-y plane.

For an elliptical disk, we use a simple rectilinear transform. The s-direction is around the ellipse, and t is along the radial direction. The transform from primitive to texture space starts by calculating the radius of the (x,y) projected point, and its azimuth:

$$r = \sqrt{x^2 + y^2}$$
$$\theta = a\tan 2(y, x)$$

(3.20)

To keep the transform simple, we used a simple rectilinear grid for the ellipse. First we form a relative vector from the center point of the ellipse to the point:

$$\vec{v} = \vec{P} - \vec{c}$$

(3.21)

We next dot this vector with unit vectors along the plane x-axis and y-axis (known from the plane construction) to get the projection distances Δx and Δy. The texture coordinates then are:

$$s = \frac{\Delta x + \sigma_x}{2 \cdot \sigma_x}$$
$$t = \frac{\Delta y + \sigma_y}{2 \cdot \sigma_y}$$

(3.22)

To transform from texture space back to 3-D space:

$$\Delta x = s \cdot 2 \cdot \sigma_x - \sigma_x$$
$$\Delta y = t \cdot 2 \cdot \sigma_{yx} - \sigma_y$$

(3.23)

Finally, the 3-D point is:

$$\vec{P} = \Delta x \cdot \hat{x} + \Delta y \cdot \hat{y} + \vec{c}$$

(3.24)

Note that with this scheme, the texture space over-fills the 3-D primitive bounded surface area. This is ok as long as we eliminate the collector points outside the ellipse through ray-tracing.

3.3 Sphere

The equation of a sphere is:

$$R^2 = x^2 + y^2 + z^2$$

(3.25)

where R is the radius of the sphere. Normally, one proceeds by substituting Equation 3.1 into 3.25 and solving for the ray-intercept distance A:

$$A^2 + A \cdot 2 \cdot (x_o l + y_o m + z_o n) + (x_o^2 + y_o^2 + z_o^2 - R^2) = 0 \qquad (3.26)$$

where we have used the normalization condition on the ray direction cosines $l^2 + m^2 + n^2 = 1$. The quadratic equation can be solved by algebraic manipulation:

$$A = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, a = 1, b = 2 \cdot (x_o l + y_o m + z_o n), c = x_o^2 + y_o^2 + z_o^2 - R^2 \qquad (3.27)$$

However, a sphere has a special geometric solution that is faster. Start by finding the distance from the ray origin to the perpendicular plane passing through the center of the sphere (distance of closest approach) by taking the dot product of ray position with ray direction cosines:

$$DCA = -(l \cdot x_o + m \cdot y_o + n \cdot z_o) \qquad (3.28)$$

If ray is going away from sphere, then the dot would be positive, and we reverse this to make DCA negative. Next we find the square distance of ray to center of sphere.

$$D^2 = x_o^2 + y_o^2 + z_o^2 \qquad (3.29)$$

If D^2 is greater than the sphere radius squared, then the ray is starting outside the sphere. If DCA is negative, then the ray does not intersect the sphere because it is going in a strictly forward direction. If DCA is positive, then the ray might hit the sphere. Assume the ray hits the sphere and next calculate half the chord length from entrance to exit points:

$$H^2 = R^2 - (D^2 - DCA^2) \qquad (3.30)$$

Clearly if this distance goes negative then the ray does not hit sphere. If the ray starts inside the sphere, then we always have two intercepts, but one is behind the starting point. We must find both the reverse entrance point and the forward exit point. Now we take the square root of the half-chord distance H and remove this distance from the total distance to the perpendicular intersection plane to get the entrance intercept distance:

$$A_{entrance} = DCA - H \qquad (3.31)$$

The exit intercept distance is gotten by adding H:

$$A_{exit} = DCA + H \qquad (3.32)$$

These solutions are valid whether the ray is inside or outside the sphere.

Given an intercept point P(x,y,z) on the sphere, we can find the outward facing surface normal by just normalizing the intercept coordinate:

$$\hat{N} = \frac{x \cdot \hat{i} + y \cdot \hat{j} + z \cdot \hat{k}}{\sqrt{x^2 + y^2 + z^2}} = \frac{\vec{P}}{R} \qquad (3.33)$$

The transform from primitive to texture space is a rectilinear to azimuth/elevation spherical polar coordinate transform. The s direction is along a circle of constant latitude (i.e., horizontal going east), and t is along a circle of constant longitude (i.e., south to north). Find the azimuth angle (0-360 degrees):

$$\theta = a \tan 2(y, x) + \pi \qquad (3.34)$$

and the elevation angle:

$$\varphi = a \cos\left(-\frac{z}{R}\right) \qquad (3.35)$$

and the texture coordinates are:

$$s = \frac{\theta}{2\pi}$$
$$t = \frac{\varphi}{\pi - \varphi_o} \qquad (3.36)$$

For the case where we are evaluating an incomplete sphere (e.g., a hemisphere), the lower bound elevation angle is φ_o.

To transform from texture space back to 3-D space:

$$\theta = s \cdot 2\pi - \pi$$
$$\varphi = t \cdot \pi + \varphi_o$$
$$x = R \cdot \cos(\theta) \cdot \sin(\varphi) \qquad (3.37)$$
$$y = R \cdot \sin(\theta) \cdot \sin(\varphi)$$
$$z = -R \cdot \cos(\varphi)$$

3.4 Cylinder

The equation of a right circular cylinder with its rotation axis aligned with the z-axis is:

$$R^2 = x^2 + y^2 \qquad (3.38)$$

where R is the radius of the cylinder. We now proceed by substituting Equation 3.1 into this equation and solving for the ray-intercept distance A:

$$A^2(l^2 + m^2) + A \cdot 2 \cdot (x_o l + y_o m) + (x_o^2 + y_o^2 - R^2) = 0 \qquad (3.39)$$

The quadratic equation can be solved by algebraic manipulation:

$$A = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, a = (l^2 + m^2), b = 2 \cdot (x_o l + y_o m), c = x_o^2 + y_o^2 - R^2 \tag{3.40}$$

If the first coefficient a is small, the ray is parallel to the rotation axis of the cylinder and will probably not hit the conic. If the ray origin is outside the cylinder (i.e., c is > 0) then there are no intercepts. If the ray is inside, then we will need to check for the ray hitting the end disk planes. We actually need to return a valid hit in this case and default the two intercept distances to large values.

Next we evaluate the radical to see if there are any real roots. If the argument of the square root is not positive, then there are no intercepts. Again, if the radical is zero, then in principle there are two degenerate roots, which we do not consider further. If the radical is positive, then:

$$A_{entrance} = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \tag{3.41}$$

$$A_{exit} = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \tag{3.42}$$

Given an intercept point P(x,y,z) on the cylinder, we can find the outward facing surface normal by just normalizing the intercept coordinate:

$$\hat{N} = \frac{x \cdot \hat{i} + y \cdot \hat{j}}{\sqrt{x^2 + y^2}} + 0\hat{k} = \frac{\vec{P}(x,y)}{R} \tag{3.43}$$

where there is no z-component to the normal.

The transform from primitive to texture space is a rectilinear to polar coordinate transform in the x-y plane. The s direction is along a circle, and t is along the cylinder z-axis. Find the azimuth angle (0-360 degrees):

$$\theta = a \tan 2(y, x) + \pi \tag{3.44}$$

and the texture coordinates are:

$$s = \frac{\theta}{2\pi}$$
$$t = \frac{z - z_{min}}{z_{max} - z_{min}} \tag{3.45}$$

To transform from texture space back to 3-D space:

$$\theta = s \cdot 2\pi - \pi$$
$$x = R \cdot \cos(\theta)$$
$$y = R \cdot \sin(\theta)$$
$$z = t \cdot (z_{\max} - z_{\min}) + z_{\min}$$

(3.46)

Three different orientations for a cylinder are used in LRST, one along the x-axis, one along the y-axis, and one along the z-axis, as described above. The equations are all the same, except one must substitute the appropriate coordinate permutation for x, y, and z.

3.5 Cone

The equation of a right circular cone with its rotation axis aligned with the z-axis is:

$$z^2 \cdot \tan^2(\delta) = x^2 + y^2 \qquad (3.47)$$

where δ is the cone apex (half) angle. We now proceed by substituting Equation 3.1 into this equation and solving for the ray-intercept distance A:

$$A^2(l^2 + m^2 - n^2 \tan^2(\delta)) + A \cdot 2 \cdot (x_o l + y_o m - z_o n \tan^2(\delta)) + (x_o^2 + y_o^2 - z_o^2 \tan^2(\delta)) = 0 \quad (3.48)$$

The quadratic equation can be solved by algebraic manipulation:

$$A = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

(3.49)

$$a = (l^2 + m^2 - n^2 \tan^2(\delta)), b = 2 \cdot (x_o l + y_o m - z_o n \tan^2(\delta)), c = x_o^2 + y_o^2 - z_o^2 \tan^2(\delta)$$

In general, this equation describes two cones, tip to tip, with touching vertices at the origin. There are three ray-intercept situations we need to consider. First the ray could be traveling generally perpendicular to the rotation axis and there will be two intercepts with the same cone. Second, the ray is headed mostly parallel to the rotation axis and there is a single intercept with the cone in the lower half space, and another intercept with the cone in the upper half space. Since we are interested in the cone located in the lower half space only, we need to reject the upper intercept and determine if the ray is entering or exiting the lower cone. Third, the ray can be traveling mostly parallel to one cone surface and intercepts only the other cone. Again we need to determine which cone is hit, and if the ray is entering or exiting that cone.

If a in Equation 3.49 is positive and greater than a small tolerance, the ray is headed such that there should be two intercepts with the same cone (case 1). We follow the same calculations as in Equations 3.41 and 3.42.

If a is less than the negative of the small tolerance, then the ray is headed along the rotation axis (case 2). If the ray is traveling along the +z-axis (i.e., l is positive), then the intercept is an exit and the distance is given by Equation 3.42. Otherwise, the intercept is an entrance and Equation 3.41 is used.

If $|a| \leq tolerance$, then the ray is traveling parallel to one cone and might intercept the other cone (case 3). We use the b coefficient to determine whether the intercept is entrance or exit. If b is negative, then the intercept is an entrance. If b is positive, then it's an exit. If b is nearly zero, then the ray either missed the cones or runs along the surface, in which case we reject any intercepts. In either case the ray-intercept distance is:

$$A = -\frac{c}{2 \cdot b}$$ (3.50)

The ray can only hit one cone, but we are not sure which one. For a solid cone we also intercept the ray with the planar end disks. We will use the knowledge of whether the cone intercept is entrance/exit and the distance to both end-planes to discriminate which cone is being hit.

Given an intercept point P(x,y,z) on the lower cone, we can find the outward facing surface normal by taking the divergence of Equation 3.47 and normalizing to unit length:

$$\hat{N} = \frac{x \cdot \hat{i} + y \cdot \hat{j} - z \tan^2(\delta)\hat{k}}{\sqrt{x^2 + y^2 + z^2 \tan^4(\delta)}}$$ (3.51)

The transform from primitive to texture space is circles around the cone at equally spaced heights. The s direction is along a circle, and t is along the cone z-axis, just like with the cylinder. Find the azimuth angle (0-360 degrees):

$$\theta = a \tan 2(y, x) + \pi$$ (3.52)

and the texture coordinates are:

$$s = \frac{\theta}{2\pi}$$
$$t = \frac{z - z_{min}}{z_{max} - z_{min}}$$ (3.53)

To transform from texture space back to 3-D space:

$$\theta = s \cdot 2\pi - \pi$$
$$x = z \cdot \tan(\delta) \cdot \cos(\theta)$$
$$y = z \cdot \tan(\delta) \cdot \sin(\theta)$$
$$z = t \cdot (z_{max} - z_{min}) + z_{min}$$ (3.54)

Three different orientations for a cone are used in LRST, one along the x-axis, one along the y-axis, and one along the z-axis, as described above. The equations are all the same, except one must substitute the appropriate coordinate permutation for x, y, and z.

3.6 Elliptical Cylinder

The equation of an elliptical cylinder with its rotation axis aligned with the z-axis is:

$$1 = \frac{x^2}{\alpha^2} + \frac{y^2}{\beta^2} \tag{3.55}$$

where α is the semi-major axis along x, and β is the semi-minor axis along y. We proceed by substituting Equation 3.1 into this equation and solving for the ray-intercept distance A:

$$A^2(\frac{l^2}{\alpha^2} + \frac{m^2}{\beta^2}) + A \cdot 2 \cdot (\frac{x_o l}{\alpha^2} + \frac{y_o m}{\beta^2}) + (\frac{x_o^2}{\alpha^2} + \frac{y_o^2}{\beta^2} - 1) = 0 \tag{3.56}$$

The quadratic equation can be solved by algebraic manipulation:

$$A = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

$$a = (\frac{l^2}{\alpha^2} + \frac{m^2}{\beta^2}), b = 2 \cdot (\frac{x_o l}{\alpha^2} + \frac{y_o m}{\beta^2}), c = \frac{x_o^2}{\alpha^2} + \frac{y_o^2}{\beta^2} - 1 \tag{3.57}$$

There are two ray-intercept situations we need to consider. First the ray could be traveling generally perpendicular to the rotation axis and there will be no intercepts with the cylinder. But, if the ray starts inside the cylinder, then it will intercept each end-plane. We determine this condition by a being very small (note that it is always positive).

Second if the ray is not parallel to the z-axis, then we test for cylinder intercepts in the usual manner, using Equations 3.41 and 3.42.

Given an intercept point P(x,y,z) on the cylinder, we can find the outward facing surface normal by taking the divergence of Equation 3.55 and normalizing to unit length:

$$\hat{N} = \frac{\frac{x}{\alpha^2} \cdot \hat{i} + \frac{y}{\beta^2} \cdot \hat{j}}{\sqrt{\frac{x^2}{\alpha^4} + \frac{y^2}{\beta^4}}} + 0\hat{k} \tag{3.58}$$

The transform from primitive to texture space is a rectilinear to polar coordinate transform in the x-y plane. The s direction is along an ellipse, and t is along the cylinder z-axis. Find the azimuth angle (0-360 degrees):

$$\theta = a \tan 2(y, x) + \pi \tag{3.59}$$

and the texture coordinates are:

$$s = \frac{\theta}{2\pi}$$

$$t = \frac{z - z_{min}}{z_{max} - z_{min}}$$

(3.60)

To transform from texture space back to 3-D space:

$$\theta = s \cdot 2\pi - \pi$$

$$r_\theta = \frac{1}{\sqrt{\dfrac{\cos^2(\theta)}{\alpha^2} + \dfrac{\sin^2(\theta)}{\beta^2}}}$$

$$x = r_\theta \cdot \cos(\theta)$$

$$y = r_\theta \cdot \sin(\theta)$$

$$z = t \cdot (z_{max} - z_{min}) + z_{min}$$

(3.61)

3.7 Elliptical Cone

The equation of an elliptical cone with its rotation axis aligned with the z-axis is:

$$\frac{z^2}{\gamma^2} = \frac{x^2}{\alpha^2} + \frac{y^2}{\beta^2}$$

(3.62)

where α is the semi-major axis along x, and β is the semi-minor axis along y, when $z=\gamma$. We proceed by substituting Equation 3.1 into this equation and solving for ray-intercept distance A:

$$A^2 \left(\frac{l^2}{\alpha^2} + \frac{m^2}{\beta^2} - \frac{n^2}{\gamma^2}\right) + A \cdot 2 \cdot \left(\frac{x_o l}{\alpha^2} + \frac{y_o m}{\beta^2} - \frac{z_o n}{\gamma^2}\right) + \left(\frac{x_o^2}{\alpha^2} + \frac{y_o^2}{\beta^2} - \frac{z_o^2}{\gamma^2}\right) = 0$$

(3.63)

The quadratic equation can be solved by algebraic manipulation:

$$A = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

$$a = \left(\frac{l^2}{\alpha^2} + \frac{m^2}{\beta^2} - \frac{n^2}{\gamma^2}\right), b = 2 \cdot \left(\frac{x_o l}{\alpha^2} + \frac{y_o m}{\beta^2} - \frac{z_o n}{\gamma^2}\right), c = \frac{x_o^2}{\alpha^2} + \frac{y_o^2}{\beta^2} - \frac{z_o^2}{\gamma^2}$$

(3.64)

The ray-intercept situations are identically the same as for a right circular cone. We therefore follow the same logic as outlined in Equations 3.41, 3.42, and 3.50.

Given an intercept point P(x,y,z) on the lower cone, we can find the outward facing surface normal by taking the divergence of Equation 3.62 and normalizing to unit length:

$$\hat{N} = \frac{\dfrac{x}{\alpha^2} \cdot \hat{i} + \dfrac{y}{\beta^2} \cdot \hat{j} - \dfrac{z}{\gamma^2} \cdot \hat{k}}{\sqrt{\dfrac{x^2}{\alpha^4} + \dfrac{y^2}{\beta^4} + \dfrac{z^2}{\gamma^4}}} \tag{3.65}$$

The transform from primitive to texture space is a rectilinear to polar coordinate transform in the x-y plane. The s direction is along an ellipse, and t is along the cone z-axis. Find the azimuth angle (0-360 degrees):

$$\theta = a \tan 2(y, x) + \pi \tag{3.66}$$

and the texture coordinates are:

$$s = \frac{\theta}{2\pi}$$
$$t = \frac{z - z_{min}}{z_{max} - z_{min}} \tag{3.67}$$

To transform from texture space back to 3-D space:

$$\theta = s \cdot 2\pi - \pi$$
$$\xi = \frac{\gamma^2}{\alpha^2 \cdot z^2}$$
$$\eta = \frac{\gamma^2}{\beta^2 \cdot z^2}$$
$$r_\theta = \frac{1}{\sqrt{\dfrac{\cos^2(\theta)}{\xi^2} + \dfrac{\sin^2(\theta)}{\eta^2}}} \tag{3.68}$$
$$x = r_\theta \cdot \cos(\theta)$$
$$y = r_\theta \cdot \sin(\theta)$$
$$z = t \cdot (z_{max} - z_{min}) + z_{min}$$

3.8 Ellipsoid

The equation of an ellipsoid is:

$$1 = \frac{x^2}{\alpha^2} + \frac{y^2}{\beta^2} + \frac{z^2}{\gamma^2} \tag{3.69}$$

where α is the semi-axis along x, β is the semi-axis along y, and γ is the semi-axis along z. We proceed by substituting Equation 3.1 into this equation and solving for the ray-intercept distance A:

$$A^2(\frac{l^2}{\alpha^2}+\frac{m^2}{\beta^2}+\frac{n^2}{\gamma^2})+A\cdot 2\cdot(\frac{x_o l}{\alpha^2}+\frac{y_o m}{\beta^2}+\frac{z_o n}{\gamma^2})+(\frac{x_o^2}{\alpha^2}+\frac{y_o^2}{\beta^2}+\frac{z_o^2}{\gamma^2})=0 \qquad (3.70)$$

The quadratic equation can be solved by algebraic manipulation:

$$A=\frac{-b\pm\sqrt{b^2-4ac}}{2a},$$

$$a=(\frac{l^2}{\alpha^2}+\frac{m^2}{\beta^2}+\frac{n^2}{\gamma^2}), b=2\cdot(\frac{x_o l}{\alpha^2}+\frac{y_o m}{\beta^2}+\frac{z_o n}{\gamma^2}), c=\frac{x_o^2}{\alpha^2}+\frac{y_o^2}{\beta^2}+\frac{z_o^2}{\gamma^2} \qquad (3.71)$$

The ray-intercept situations are the same as for a sphere. We therefore follow the same logic as outlined in Equations 3.41, 3.42.

Given an intercept point P(x,y,z) on the ellipsoid, we can find the outward facing surface normal by taking the divergence of Equation 3.69 and normalizing to unit length:

$$\hat{N}=\frac{\frac{x}{\alpha^2}\cdot\hat{i}+\frac{y}{\beta^2}\cdot\hat{j}+\frac{z}{\gamma^2}\cdot\hat{k}}{\sqrt{\frac{x^2}{\alpha^4}+\frac{y^2}{\beta^4}+\frac{z^2}{\gamma^4}}} \qquad (3.72)$$

The transform from primitive to texture space is a rectilinear to polar coordinate transform in the x-y plane. The s direction is along an ellipse, and t is along the z-axis. Find the azimuth angle (0-360 degrees):

$$\theta=a\tan 2(y,x)+\pi \qquad (3.73)$$

and the texture coordinates are:

$$s=\frac{\theta}{2\pi}$$

$$t=\frac{z-z_{min}}{z_{max}-z_{min}} \qquad (3.74)$$

To transform from texture space back to 3-D space:

$$\theta = s \cdot 2\pi - \pi$$

$$\xi = \frac{1}{\alpha^2 \cdot (1 - \frac{z^2}{\gamma^2})}$$

$$\eta = \frac{1}{\beta^2 \cdot (1 - \frac{z^2}{\gamma^2})}$$

$$r_\theta = \frac{1}{\sqrt{\frac{\cos^2(\theta)}{\xi^2} + \frac{\sin^2(\theta)}{\eta^2}}} \quad (3.75)$$

$$x = r_\theta \cdot \cos(\theta)$$

$$y = r_\theta \cdot \sin(\theta)$$

$$z = t \cdot (z_{max} - z_{min}) + z_{min}$$

This completes the discussion of primitive ray-intercept calculations. In general, all the conic sections are expressed in a canonical coordinate system. But, the model is built using a hierarchical system of affine transforms. Each solid therefore has its own target model body frame to local solid frame transform assigned to it. Internally this transform is represented as a 3 by 5 matrix. The forward transform takes a ray from the body frame to the local solid frame by:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} \quad (3.76)$$

where the a_{ij} matrix elements perform a rotation on the input vector $\bar{x}$, and the right hand column matrix is the offsets to the rotated vector. There are 3x3=9 rotation elements, 3 offset components for the forward transform, and 3 offsets for the backward transform (totaling 15 elements, hence a 3x5). A ray that is to be traced to a solid is first transformed into the canonical frame of the solid and then traced using the equations above. The ray origin is transformed using Equation 3.76. The ray direction cosines are transformed using just the 3x3 rotation part of the transform.

Only two solids are completely enclosed by their own surface definition, the sphere and the ellipsoid. In all the other cases, the solid is completed by one or two clipping planes. To properly close a solid, all normal vectors must point outward.

As we trace each primitive face (conic or plane), we keep track of entrance and exit intercept distances. If a ray trace either increases an entrance distance, or decreases an exit distance, we keep those intercepts. If the exit distance ever becomes less than the entrance distance, then we have missed the solid. This logic only works for convex hull solids, which all these solids are.

As we trace the solids within a single CSG solid, we keep track of all the entrance and exit intercepts, their distances, and the respective primitive pointers. Once all the primitives of a CSG solid have been traced, we merge/sort the entrance/exit points to find the higher level (abstract) solid along the shot-line of the ray being traced. We then only consider the first entrance point forward of the ray starting origin, since it is the closest intercept. Once the distance to the closest intercept is known, the actual intercept coordinates can be determined using Equation 3.1, by propagating the ray forward by the intercept distance.

For simple target models with just a couple of primitive solids, the ray tracer could simply loop through all the solids and test the ray against them one at a time. But, when the number of primitives becomes large (perhaps more than 10), then it becomes necessary to employ some form of ray-trace acceleration. The basic technique is usually to spatially partition space into volumes that contain just a few primitives localized in that region. This way the ray-tracer avoids doing ray-intercept tests on most of the primitives.

In LRST we employ a classic technique called Binary Spatial Partitioning, or BSP. We first parse the model file and construct all the primitives in the model. Then each primitive is bounded by an axis-aligned box. At the top level, a single bounding box is put around all primitives (including the translations and rotations). This is the root or parent BSP cell. Starting with this cell, child cells are generated by cutting the parent cell in half using a plane that is parallel to one of the principal planes (i.e., x-y plane, x-z plane, or y-z plane). We start the binary splitting process by deciding on where the first splitting plane is to be put in the parent X-BSP cell. The plane is put somewhere between the ends of the box. The best approach seems to be to place the splitting plane where half the member solids are on one side and the other half on the other side. Next, the two child cells are split along the *next* direction as 2 child Y-BSP cells. The splitting plane is placed such that half the members are on one side, and so forth until there is a minimum number of members in a descendent cell. The algorithm is recursive, and the stopping criteria is either at a minimum number of members or when further bi-section does not reduce the member count.

With the BSP built, rays can be traced against it and the member solids. First the ray is tested recursively against the BSP cells down the family tree until a leaf cell is encountered with member solids to be traced. The traversal algorithm always works to test the ray against the nearest BSP cell or its contents such that if a valid solid ray-intercept is found, then the traversal can be terminated.

For LRST, we are using the ray tracing mechanics to test for target assembly obstructions. Thus we start a ray at a collector point, point it toward either the laser source or the reflection direction, and do the ray trace. If the ray encounters no valid intercepts with any solids before leaving the BSP top grandparent cell, then there is no LOS obstruction. If the local surface normal at a collector point is pointing away from the ray (i.e., the dot product between the two vectors is negative), then there is no need to trace any ray because the collector point must be obstructed by the same solid (i.e., the back facing normal is culled).

The order of ray tracing is first to test if the laser beam can illuminate the collector point. If it can, then we do a second test to see if any specularly reflected light can escape without being

scattered yet again by other model components. If the incident laser beam has direction cosines $\hat{s}_o$, then the reflected ray LOS $\hat{s}_1$ is given by Snell's Law for reflection:

$$\hat{s}_1 = \hat{s}_o - 2 \cdot (\hat{s}_o \circ \hat{N})\hat{N} \qquad (3.77)$$

4.0 Radiometric Modeling

At the heart of determining hazards is predicting scattered intensities at observer positions as a function of time. In Figure 1.1 we presented the basic algorithmic flow in LRST from photon emission at the laser device, propagation of the laser beam to missile target, reflections into the spherical volume surrounding the target, propagation to observers through the atmosphere, and finally hazard evaluation at the observer. Figure 4.1 shows the same basic flow. In this section we will proceed down this block diagram and discuss how each step is modeled.

```
                   ┌─────────────────────┐
                   │    Laser Beam       │
                   │ Propagation to Target│
                   └─────────────────────┘
                              ↓
                       ┌─────────────────────┐
                       │  Irradiance Profile │
                       │   at Target plane   │
                       └─────────────────────┘
                              ↓
             ┌─────────────────────┐
             │  Geometry, BRDFs    │
             │  at collector points│
             └─────────────────────┘
                              ↓
                   ┌─────────────────────┐
                   │Scattered Cross Sections│
                   │     at Sphere       │
                   └─────────────────────┘
                              ↓
           ┌─────────────────────┐
           │  Range, Dynamics,   │
           │Atmospheric Attenuation│
           └─────────────────────┘
                              ↓
                 ┌─────────────────────┐
                 │Intensity Time-Histories│
                 │    at Observers     │
                 └─────────────────────┘
                              ↓
         ┌─────────────────────┐
         │     ANSI Z136       │
         └─────────────────────┘
                              ↓
                 ┌─────────────────────┐
                 │      Hazards        │
                 └─────────────────────┘
```
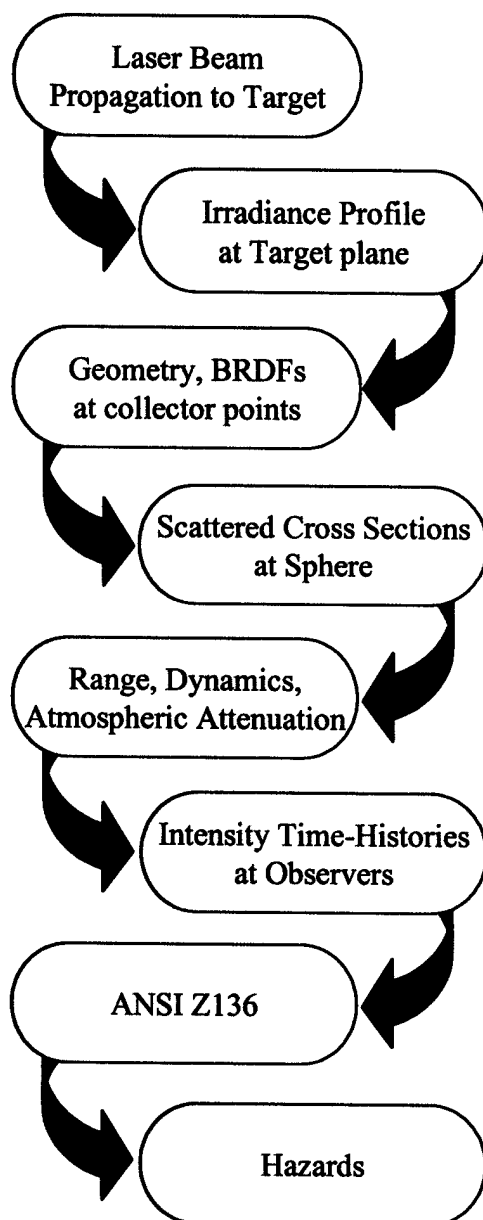
Figure 4.1. Basic Computational Flow for Radiometrics Down to Hazards

4.1 Laser Propagation

While LRST should ultimately be capable of modeling any generic laser system, for initial versions at a minimum, it should include the four ABL lasers, which are the Laser Ranger (11.15 μm), the Track Illuminator (TILL, 1.03 μm), the Beacon Illuminator (BILL, 1.06 μm), and the High Energy Laser (HEL, 1.315 μm). We are interested in a model that predicts both the peak intensity on the optical axis, and its full intensity profile off the axis.

Propagation of laser beams through the atmosphere is a complicated wave-optic process that demands significant computational work if a high fidelity far field point spread function (PSF) estimate is required. The far field is typically very dynamic, stochastic, dependent on the physical state of the atmosphere along the propagation path, and controllable to some extent with adaptive optics and tilt loops within the ABL. A number of propagation models have been developed and are being used by various laser programs. However, for the initial version of LRST, we really only require a reasonable and *conservative* estimate of the far field intensity distribution that is simple to use and understand.

There are two basic pupil intensity configurations that are commonly encountered: a uniformly filled pupil and a Gaussian intensity distribution across the pupil. The uniformly filled (or Flat Top) pupil is how most high energy lasers fill the pupil area. Laser illuminators on the other hand typically exhibit a Gaussian intensity profile. We will present equations for the far field intensity profiles for these two cases, even though we understand that at any instantaneous moment the actual profile at the target plane is probably highly scintillated. Our approximation of using a constant conservative intensity profile and not a high fidelity and highly dynamic beam profile is designed to minimize computation and preserve conservatism. LRST is designed to evaluate the existence of eye-hazards and not predict system performance. We therefore are assuming that the time-average profile will always be less than the profile represented by the diffraction limit. Both lower order and higher order wavefront aberrations will tend to drive energy out of the main beam and thus uphold this assumption.

This is not to say the irradiance at the target plane is constant. The irradiance scales as $1/range^2$, assumes the ABL can maintain a constant beam divergence angle. The irradiance also suffers atmospheric extinction, which scales exponentially with range. We will explicitly ignore tilt jitter of the beam on target. We justify this either by claiming our diffraction limited beam profile is more conservative, or allowing the user to include expected residual system pointing jitter in the divergence specification of the beam. The latter is probably preferred.

The ABL concept of using multiple lasers to illuminate a target means that beams of different size are placed on the target at different locations, usually nearer the nose. We will need to provide multiple body fixed aimpoint coordinates for these beams to nominally point at during the simulation. We also realize that a real system might have problems with boresight drift or other hardware/geometric issues that preclude keeping the beams absolutely fixed to a single point. In fact, one might argue that the ABL might not point very well at all, and hence the laser beams could land (dance) anywhere on the target. As an option we allow the user to specify the beam profile as a flood-load, or uniform intensity profile at the target plane. The user specifies a nominal Gaussian beam profile, but the code calculates the peak Gaussian irradiance and assigns

this single value to the uniform distribution. This obviously creates energy where there is none. The degree to which energy is created depends on the size of the target relative to the original Gaussian size. Thus, if the nominal Gaussian beam is 1 m², and the target size is 1 by 10 meters, then there is potentially a gain of 10 in the beam energy on target. This clearly is a conservative approach to ensuring that the beam can not wander onto parts of the target that might otherwise reflect more energy to an observer because of different target structure.

According to Born and Wolf [2], a perfectly corrected flat-top pupil propagates in a vacuum such that the diffraction-limited far field (in the Fraunhofer zone, i.e., near focus) intensity (sometimes called an Airy pattern) is given by:

$$I(r,z) = \frac{I_o}{\left(1 - \varepsilon^2\right)^2} \cdot \left[\frac{2 \cdot J_1(kaw)}{kaw} - \varepsilon^2 \frac{2 \cdot J_1(k\varepsilon aw)}{k\varepsilon aw}\right]^2 \qquad (\text{W/m}^2) \qquad (4.1)$$

where:

ε obscuration ratio, secondary mirror to primary mirror diameter

$J_1(*)$ Bessel Function of the first kind

k wave number $= 2\pi/\lambda$

λ laser wavelength

a pupil (e.g., primary mirror) radius

w observation point angular radius (i.e., $w = r/z$) - circularly symmetric beam

r $= \sqrt{x^2 + y^2}$ is the radial distance from the optical axis to the observation point

z propagation distance from the exit pupil to the observation point (i.e., range)

and where the peak on the axis intensity I_o is given by:

$$I_o = \frac{P_{laser} \cdot A_p}{\lambda^2 z^2} \qquad \text{W/m}^2 \qquad (4.2)$$

in addition:

P_{laser} is the output laser power at the exit aperture (after all system losses)

A_p is the clear pupil area (i.e., $A_p = \pi \cdot a^2 \cdot (1 - \varepsilon^2)$)

Equation 4.2 holds for *any* shaped aperture with a uniform intensity profile. The shape of the far field is driven primarily by wavelength and aperture diameter. For a circular unobscured aperture the far field PSF diameter is:

$$\theta_{diam} = \frac{2.44 \cdot \lambda}{D} \qquad (\text{rad}) \qquad (4.3)$$

where D is the pupil diameter (i.e., D=2a), and θ_{diam} is defined as the angular distance between the first zeros in Equation 4.1. The actual far field PSF pattern extends to infinity in the perpendicular directions, but more than 95% of the energy resides close the optical axis and

within the radius of the first three rings. Central obscurations (e.g., a secondary mirror in a Cassegrain telescope) tend to narrow the main lobe and raise the secondary ring lobes. But according to Equation 4.2, for the same power-area product the peak far field intensity remains constant.

The flat-top beam is uniquely defined by four parameters, the exit pupil diameter (or radius), the wavelength, the central obscuration ratio, and the output laser power.

According to Yariv [3], a Gaussian beam propagates in a vacuum such that the electric field distribution E(r,z) is:

$$E(r,z) = E_o \cdot \frac{\omega_o}{\omega_z} \cdot \exp\left[-i(k \cdot z - \eta_z) - r^2\left(\frac{1}{\omega_z^2} + \frac{ik}{2R_z}\right)\right] \quad (4.4)$$

where,

z   distance from the beam waist (narrowest point) to the observation point (i.e., range)

ω_o beam waist (narrowest point) radius (i.e., radius where the intensity is down by e^{-2})

ω_z beam radius at a range z (i.e., radius where the intensity is down by e^{-2} in the far field)

i imaginary number (i.e., $\sqrt{-1}$)

k   wave number $= 2\pi/\lambda$

η_z $= \tan^{-1}\left(z/\pi\omega_o^2\right)$

R_z $= z \cdot \left[1 + \left(\frac{\pi\omega_o^2}{\lambda z}\right)^2\right]$ the phase front radius of curvature for large z

The beam size at a distance z from the waist is:

$$\omega_z = \omega_o \cdot \sqrt{1 + \left(\frac{\lambda z}{\pi\omega_o^2}\right)^2} \quad \text{(m)} \quad (4.5)$$

which only grows in size as z increases. Close in to the beam waist, ω_z grows more slowly. When $\frac{\lambda z}{\pi\omega_o^2} \rangle\rangle 1$, ω_z grows linearly in range z, hence the Gaussian beam size grows linearly with a divergence angle of:

$$\theta_{div} = \frac{\omega_z}{z} \approx \frac{\lambda}{\pi\omega_o} \qquad \text{(rad)} \qquad (4.6)$$

The far field intensity is given by the magnitude squared of Equation 4.4:

$$I(r,z) = \frac{2 \cdot P_{laser}}{\pi\omega_z^2} \cdot \exp\left(-2 \cdot \frac{r^2}{\omega_z^2}\right) \qquad \text{(W/m}^2\text{)} \qquad (4.7)$$

The Gaussian beam is uniquely specified by four primary parameters, the divergence angle θ_{div}, the total average output laser power P_{laser}, the wavelength λ, and the location of the beam waist. If divergence is used as a user input, Equation 4.6 can be used to solve for the beam waist that is then used in Equation 4.5 to solve for ω_z. Or, the beam waist ω_o can be specified directly and Equation 4.5 is used to solve for ω_z. Another popular method of specifying the Gaussian divergence is via the half-width/half-maximum (HWHM) angle which is related to ω_z by:

$$\omega_z = 1.69864 \cdot z \cdot \theta_{HWHM} \qquad \text{(m)} \qquad (4.8)$$

which, like divergence, only applies to the far field.

The location of the beam waist can actually be a little ambiguous. In Equations 4.4, 4.5, and 4.7 it is always assumed to be located at the exit pupil of the optics. When $\frac{\lambda z}{\pi \omega_o^2}\rangle\rangle 1$, then Equation 4.6 can be used to solve for the beam waist ω_o, which in turn can be used to solve for the far field spot size ω_z. From Equation 4.5 it is clear that $\omega_z \geq \omega_o$. When the far field spot size approaches the beam waist size $\omega_z \approx \omega_o$, then the target is effectively not in the far field. When the divergence is such that the far field spot is smaller than the waist, then the focus plane (i.e., where the waist is located and z=0) is no longer located at the exit pupil, but rather it is somewhere in between the transmitter and the target, or even beyond the target.

For our code perspective, the location of the waist is really not terribly important. It will be assumed that the system optics has applied the appropriate focus term to the Gaussian wavefront to achieve the desired spot size at the target plane. Therefore the beam divergence will be used to solve for the far field spot size directly using $\omega_z = \theta_{div} \cdot z$. The beam waist is assumed to be the right size and located in the right place to achieve the desired spot size ω_z.

The peak on the axis intensity of a Gaussian beam with a given total output laser power will have twice the intensity as the equivalent sized flat-top beam propagation. The far field spot size for a Gaussian beam is:

$$\omega_z = \frac{\lambda \cdot z}{\pi\omega_o} \qquad \text{(m)} \qquad (4.9)$$

substituting this into Equation 4.7 for the peak on-axis intensity gives:

$$I_{oo} = 2 \cdot \frac{P_{laser} \cdot \pi \omega_o^2}{\lambda^2 z^2} = 2 \cdot \frac{P_{laser}}{\pi \omega_z^2} \qquad (\text{W/m}^2) \qquad (4.10)$$

If we associate πw_o^2 with the equivalent pupil area of a flat-top beam, then QED. A Gaussian apodized pupil is more efficient at putting its energy near the optical axis.

The next dimension we need to consider is time. Lasers can be CW (continuous wave), or pulsed. While real world pulse shapes can be complicated, we will limit our model to either CW or a simple square wave with a pulse repetition frequency (PRF).
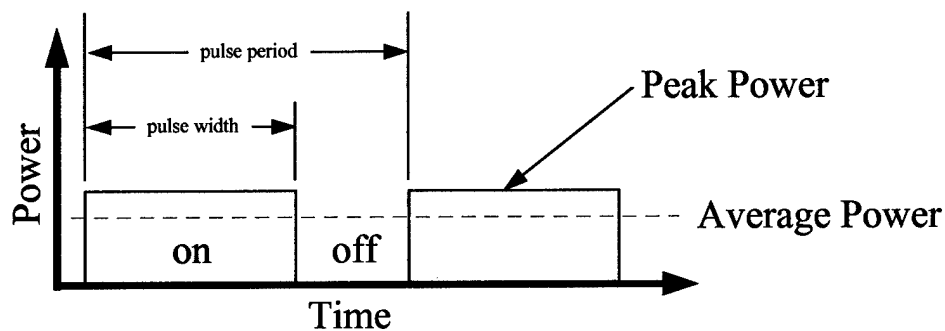
Figure 4.2. Notional Laser Pulse Train

In Figure 4.2 a laser pulse train is shown schematically. In the preceding equations where laser power was used, we need to specify either peak or average power. To avoid confusion we will chose *peak power* for our equations. The average power is related to peak power simply by:

$$P_{average} = P_{peak} \cdot \frac{\Delta t_{on}}{\Delta t_{on} + \Delta t_{off}} = P_{peak} \cdot \Delta t_{on} \cdot PRF \qquad (4.11)$$

A CW laser has an infinite on-time and a zero repetition rate, such that the product is always 1, and the average and peak powers are the same.

Therefore to specify the temporal characteristics of a laser device we will need Peak power, the pulse width in seconds, and the PRF as a frequency (i.e., pulses per second). Clearly an alternative is to refer to the energy per pulse ($E_{pulse} = P_{peak} \cdot t_{pulse-width}$). However, if we did this then we would be mixing power units for a CW laser, and energy units for a pulse laser, which would eventually lead to misunderstandings and input errors. The distinction between a CW laser and a pulsed laser becomes most pronounced when the laser on-time is short compared to the pulse-to-pulse period.

While Equation 4.4 gives the profile of an ideal Gaussian beam at any range, we will use the simpler form in Equation 4.7 that assumes the target is always in the far field of some laser

transmitter. Adding the atmospheric extinction from transmitter to target plane, we model the target plane peak irradiance as:

$$I(r,z) = \tau_{atm} \frac{2 \cdot P_{laser}}{\pi \omega_z^2} \cdot \exp\left(-2 \cdot \frac{r^2}{\omega_z^2}\right) \qquad (\text{W/m}^2) \qquad (4.12)$$

where the beam radius ω_z at range is:

$$\omega_z = \theta_{div} \cdot z \qquad (\text{m}) \qquad (4.13)$$

Notice it is possible to pick a divergence angle that is small enough to be unphysical given the characteristics of the ABL. When we use a flood-load profile, r=0 and the constant peak irradiance is:

$$I(r,z) = \tau_{atm} \frac{2 \cdot P_{laser}}{\pi \omega_z^2} \qquad (\text{W/m}^2) \qquad (4.14)$$

4.2 Radiometric Scattering

One of the primary objectives of the LRST modeling is to track reflected intensity distributions in three dimensions, regardless of how those intensities are estimated. In other words, the code needs a mechanism for bookkeeping where the scattered energy goes in *all* directions, for all times. The emphasis here is to discuss the code implementation for doing this bookkeeping as it relates to capturing the reflected intensities and projecting the results onto objects in the simulation scene.

There are a few basic requirements that any technique must meet. The scattered intensities need to be captured in memory in such a way that an intensity can be predicted for any point in three dimensional space. There are a number of different kinds of observing objects within the simulation at whose position we want to be able to predict the reflected intensity. In general we really do not require knowledge of all intensities for all space simply because most of space is empty and therefore uninteresting. But, it would be good to have the capability to insert observers anywhere into the scene, at arbitrary positions, and be able to predict the irradiant intensity there.

There are several methods we could use to scatter intensities to observation points. One method might be to follow the natural path of the incident light through the scattering process onto where the light might land in the scene. This approach has two difficulties. First we would need to follow a large number of directions from target to scene and somehow monitor the reflected intensities somehow. Second, if we did not follow a reflection direction exactly toward an observer, then we would probably miss the observer and therefore suffer severe aliasing artifacts.

Another approach would be to work backwards from the point of view of each observer. At each time step, we could reflect the laser beam off the entire target model (i.e., integrate over the target model surface area) and evaluate the instantaneous intensity at the observer point. This is essentially rendering an image of the target. While this seems to offer a high fidelity solution, it

too suffers. First, the number of computations done for just one observation point is very large. Multiply this by many time steps and a very large number of observers (spaced close enough to meet the ½ degree resolution requirement), and the approach takes far too much time to calculate. Like the first approach it too suffers from severe aliasing artifacts between time steps where a glint could pass over an observer without being detected.

The approach we use involves the best of each of these two, but adds a monitoring surface construct that guarantees that we will not miss fast dynamic glint motions sweeping across observers. The classic problem with modeling radiometrics on digital computers is some form of spatial or temporal aliasing. We need to strive to minimize all aliasing artifacts from this application because it is so important that we not miss a true hazard. We have set a requirement to instrument the scene with observers spaced no more than ½ degree apart, as seen from the target point. This localizes hazards well enough spatially, while not over-sampling the problem too much to cause an unnecessary computational burden. Most bi-directional reflectivity distribution functions (BRDF) do not scatter light into a cone with a angle smaller than this. Thus there is a good match between adequate sample spacing and realistic scattering. When a specular reflection does occur with an angular width less than a half degree, we will need to ensure that a conservative method is used to account for regions near the peak. The method needs to smoothly span geometric discontinuities such as where the line of sight to the earth and space meet. Temporally we need to worry about dynamic motions of the laser source, target, and observers. Specifically it is possible for very rapid glint events to occur (perhaps on the millisecond time scale?) which are quite important to the vulnerability of observers. Since intensity is not the measure of vulnerability, but rather the time history of intensity, the technique must lend itself to predicting the full time history of intensity at the observer point right down to the most rapid glint event.

Finally, the code implementation needs to be practical. The memory requirements to host the intensity distributions must be manageable. The use of a disk to store intermediate results is always reasonable, but very large arrays will likely lead to disk-swapping penalties. Access speed is also a consideration. The code needs to be able to read and write intensity values quickly. In this implementation we probably should not use the disk for intermediate results.

Between the target trajectory point and any observer point we surround the target with an optical cross section sphere (a.k.a., the Golden Sphere). A basic starting premise for the reflection geometry is that all observers (i.e., all interesting points in space where an intensity is desired) shall be located a substantial distance away from the target. For example, a typical theater ballistic missile (TBM) would be approximately 15 km above mean sea level before the ABL would engage it. The closest observer is probably someone on the ground at least 15 km away. A 10-m tall TBM target then would subtend an angle of 0.7 milliradian (about the diffraction limit for the eye at this wavelength), or less, for all observers. As far as an observing eye can tell, the target is essentially an unresolved radiant point at the origin of the target model. For an observer closer than this (e.g., a pilot flying closer to the target), the illuminated target image might become a resolved image on the retina (if the light gets there). Resolved or not, we will use the ANSI standard damage criteria for an unresolved point.

This means TBMs scatter light to an observer essentially along the LOS connecting the observer position and the target position. We made the approximation above, that the laser source irradiance is a collimated beam incident along a single direction (defined by a unit vector). We will also make the approximation that all reflected intensities from every part of the target surface travel from the target body origin along a single direction (again a second unit vector to any observation point). This quasi-collimated assumption allows the code to reference a total scattered intensity against a single observation LOS defined by a single set of direction cosines (unit vector).
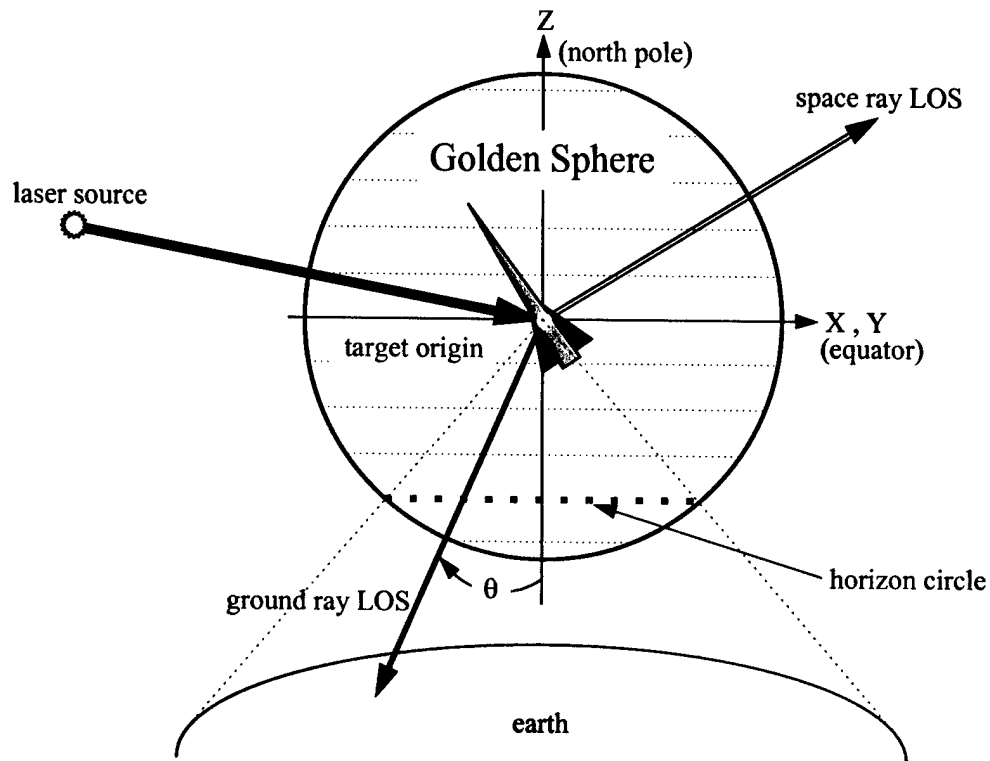
Figure 4.3. Golden Sphere Surrounding Target

Figure 4.3 shows a typical geometry for the laser source, target, earth, and the cross section sphere. The target size is exaggerated for clarity. The reflected ground or space rays represent the scattered collimated light going in a single direction through the scene. Each cross section sphere is centered on the instantaneous origin (i.e., trajectory point) of the target model at each time step in the simulation. Thus it necessarily moves with each time step, and we therefore must keep in memory the collection of spheres calculated at each time step. The poles are always aligned with the zenith direction, the negative z-axis always passing through the earth origin. The x-axis should nominally point north, while the y-axis forms a right handed coordinate frame (i.e., it points westward).

The sphere changes its attitude with time as the target moves, but only very slowly since its origin is located at the target trajectory position, which is a long distance away from the ECI origin. The sphere could have been always aligned with a celestial sphere (i.e., aligned with the

ECI *ijk* axes), but the zenith aligned geometry allows a simple circle at a single polar angle to divide the space from the ground rays. The polar angle (we should reserve the use of the term *latitude* for elliptical earth angles) of the horizon circle measured up from the -z axis is:

$$\theta \approx \sin^{-1}(\frac{R_e}{R_e + h}) \tag{4.15}$$

where R_e is the local earth radius, and h is the instantaneous altitude of a target above mean sea level. For example, for a target altitude of 15 km, the earth subtends an angle slightly greater than 86 degrees, which is almost the full lower hemisphere. Actually the dividing line is not a circle because the earth is not spherical, but the distinction is not important since the cross section sphere will cleanly make the transition between the two when it comes to intensities.

The cross section sphere is just a convenient coordinate system with which to register all other objects in the scene. It is not a physical realizable object in the scene. It never needs to be explicitly rendered or visualized. It is just a useful mathematical construct we will use for managing intermediate scattered intensities. Relative to the cross section sphere coordinate frame, the laser source direction (i.e., its position), the target attitude (orientation), and the directions to all observers (i.e., their positions), whether fixed to the ground or not, will change with time. The LOS trajectories across the surface of the cross section sphere and the target attitude rotations will all contribute to reflected spot motions in the scene, and hence how we derive the time-history intensities at all observers.

Light is incident on the target model along the LOS from laser source to target. In principle, light is then reflected from all exposed surface areas of the target along an infinite number of directions. Within the digital constraints of the computer we can only keep track of a finite number of reflection directions. Hence, the surface of the cross section sphere is meshed with a large number of finite sized pixels, or array elements, each representing the maximum reflected cross section (later to become an intensity) from the whole of the target surface, emanating along the LOS from the sphere's origin through the face of each pixel. Any observer LOS vector that pierces anywhere within the bounding edges of such a pixel (a small pyramid) is assumed to experience the same cross section (later, a radiant intensity at a given range). In Figure 4.4 we draw a single pixel on the cross section sphere with an observer looking back at the target point. Each pixel represents the worst-case cross section (intensity at range) over that small subtended solid angle. Thus all viewing directions that look anywhere through this pixel will see the same target cross section. Adjacent pixels will have different values, but we do not attempt to smooth or interpolate values while traversing the sphere's surface.
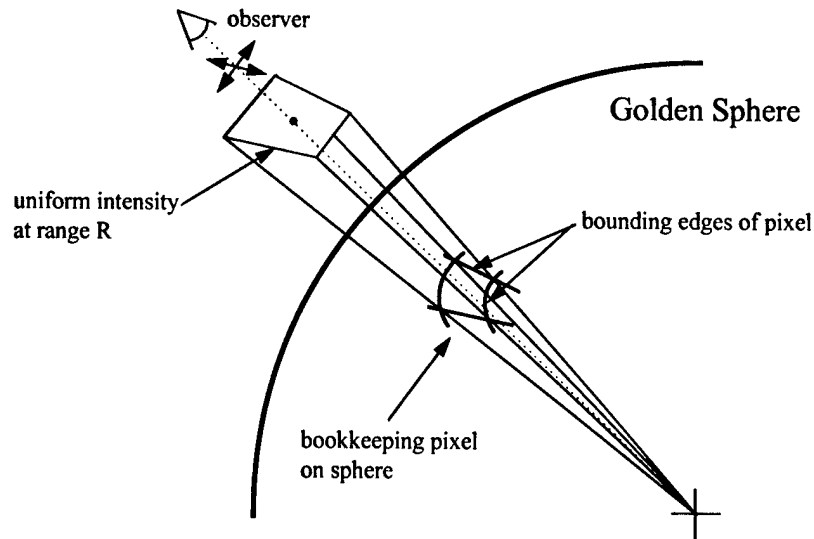
***Figure 4.4. Sphere Pixel Representation and How the Viewer Sees
a Constant Cross Section Over That Pixel***

How should the sphere pixels be allocated? The simple approach would be to use conventional spherical coordinates. But the obvious problem here is non-uniform spatial (angular) sampling near the poles. The pixels there get smaller (i.e., subtend a smaller solid angle), and they turn into narrow wedges right at the poles. The orientation of the cross section sphere puts the south pole down, probably where the closest ground observers would exist. While over-sampling along this direction would not be bad, it needlessly wastes memory and computation time.

To save memory, and computational time, we will use a slight variation on this conventional grid pattern. In elevation (i.e., the latitudes), the sphere is subdivided into equal angles, as always. This fits well with the fact that the earth horizon is at a constant elevation angle. The south pole is set to zero degrees elevation, and the north is 180 degrees. However, for each row of pixels around the sphere (i.e., at constant elevation), the number of allocated pixels varies. The meshing scheme therefore does not produce an array pattern that can be unwrapped into a simple 2-D array of cells. Figure 4.5 shows half the cross section sphere meshed in this fashion (the other half was omitted for clarity). In this figure we have exaggerated the size of the pixels. All pixel edges line up at the zero degree longitude, but nowhere else. This approach allocates enough pixels to sample the full 4π directional space with nearly equal resolution.
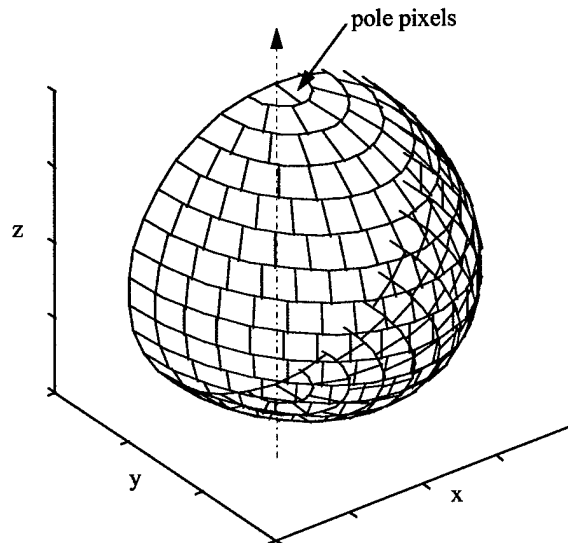
Figure 4.5. Cross Section Sphere Meshed With Equal Solid Angle Pixels

Recall that all intensities radiate from the origin of the sphere. As seen from the origin, one would want bookkeeping pixels of about equal solid angle to achieve nearly uniform spatial sampling. Not only would the pixels subtend an equal solid angle, but also they would be reasonably square (i.e., equal angle along any two orthogonal axes) over the entire sphere surface. Only near the poles do the pixels around the last row have a triangular shape, which nevertheless still have approximately the same solid angle as all the other pixels. Figure 4.6 plots the number of required pixels along a row as a function of polar angle to meet the half degree resolution requirement. A total of 165,012 pixels cover the sphere. The first rows at the poles have three pixels, and the equatorial row has 722 pixels.
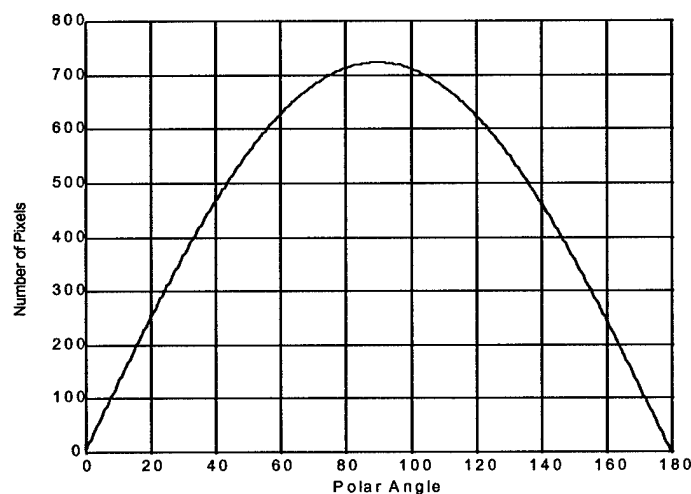
Figure 4.6. Variation in the Number Of Pixels Along a Row With Polar Angle

If $\Delta\sigma$ is the required spatial (angular) resolution, then $N_{row} = n$ int$(180/\Delta\sigma)$ rows of pixels are required, each row having:

$$N_{pixel}(i) = \max\left[1, n\,\text{int}\left(\frac{2\pi}{d\Omega/(\cos(\varphi_i) - \cos(\varphi_{i+1}))}\right)\right] \qquad (4.16)$$

pixels allocated, starting at a zero azimuth. The subscript i corresponds to the ith row of pixels, where φ_i is the lower elevation angle of the row, and φ_{i+1} is the upper elevation angle. $d\Omega = \Delta\sigma^2$ is the approximate solid angle of every pixel in the array.

The coordinate mechanics for addressing a pixel on the sphere needs to be discussed. We will need to populate pixels with cross sections, look-up values based on LOS ray direction cosines, and estimate observer and spot motions on the sphere's surface.

First we need to define how to address a pixel based on the LOS direction cosines of a unit vector pointing at something. Any point on the sphere can be fundamentally located by two angles, an azimuth angle (0 to 360 degrees, starting at x=0 and going counter-clockwise toward +y), and a polar declination angle (-90 to +90 degrees, starting at -z and going toward +z). Since the array allocation will be done by getting dynamic memory row by row, we must first determine which row the ray belongs to. The z-component of the unit vector direction cosines is used first to determine the declination angle:

$$\varphi = \sin^{-1}(z) \qquad\qquad -90 \le \varphi \le 90 \qquad (4.17)$$

The right ascension of the ray is done using an arctangent:

$$\theta = \text{mod}\left(a\tan 2(y,x) + 360, 360\right) \qquad 0 \le \theta \le 360 \qquad (4.18)$$

The row (zero based indexing) of the pixel then is:

$$row = \text{int}\left[\frac{\varphi + 90}{\Delta\sigma}\right] \qquad (4.19)$$

The column within that row is:

$$col = \text{int}\left[\frac{\theta}{2\pi/N_{pixel}(row)}\right] \qquad (4.20)$$

where the right ascension increment depends on the number of pixels in a row. Once the (row,column) of the pixel is known, one can index into the 1-D array of floating point values to either read or write cross section values:

$$index = col + Offset(row) \qquad (4.21)$$

where the Offset(*) is an array of integers which represents the running total of pixels up to the current row.

The same simple mechanics almost work in reverse, where we know the indices of a pixel and want back the polar angles. Almost, because we can only return the angles to the *center* of the pixel

$$\theta = (col + 0.5) \cdot \Delta\theta \qquad\qquad 0 \le \theta \le 360 \qquad\qquad (4.22)$$

$$\varphi = \frac{\varphi_i + \varphi_{i+1}}{2} \qquad\qquad -90 \le \varphi \le 90 \qquad\qquad (4.23)$$

where ϕ_i is the lower declination angle of the ith row. Given a right ascension,declination angle, one can immediately calculate the pointing direction cosines:

$$
\begin{aligned}
x &= \cos(\varphi) \cdot \cos(\theta) \\
y &= \cos(\varphi) \cdot \sin(\theta) \\
z &= \sin(\varphi)
\end{aligned}
\qquad\qquad (4.24)
$$

Equations 4.22 and 4.23 give the pointing direction to the center of a sphere pixel. We will also want to know the directions of each of the four corners of a pixel. In this case either the lower or upper declination angle is used, and the right ascension angle is either one edge or the other (i.e., the 0.5 faction in Equation 4.22 is either 0 or 1).

Next we turn to filling the cross section sphere with radiometric values. The first thing to understand is that a cross section sphere is populated for each time step of the engagement. The engagement geometry is first determined (frozen), the laser beam is placed on the target, and then the cross section sphere is painted with reflected values. At the next time step the process is repeated for a new sphere. The user must ensure that the scenario dynamics between time steps does not change enough to invalidate some of the assumptions we will need to make later about traversing across the sphere. The time step is typically about one second on the simulation time line.

At each laser wavelength we need to generate a unique cross section sphere. The cross sections are different at different wavelengths because of BRDF changes, and perhaps because of laser beam profile differences. The geometry must, however, be identical.

4.3 The BRDF Models

A critical capability of the Laser Range Safety Tool (LRST) simulation is accurately modeling and predicting power distributions reflecting from target missile surfaces. At optical (visible and infrared) wavelengths, the classical radiance equations for radiometry are driven primarily by microscopic geometric effects. One normally expresses the scattering equations using the concept of the bi-directional reflectivity distribution function (BRDF), which captures all the detailed scattering physics for how light interacts, whether in reflection or transmission, with a complex material surface structure. We will discuss three BRDF modeling representations used in the

LRST. The first was developed at the Environmental Research Institute of Michigan (ERIM) by J.R. Maxwell and J. Beard [5]. This will be called the Maxwell-Beard (MB) BRDF. Two other empirical analytic BRDF models will be presented for completeness. The Phong BRDF model [6] is a simplified but venerable empirical approach used for years in computer graphics. Finally, the Gaussian BRDF model is a variation on several similar useful BRDF models for doing closed-form analytic analyses. These last two models are essentially two-parameter models with only very limited capability to capture otherwise complex scattering physics.

These three are the currently accepted standard models being used by AFRL for several simulation efforts. These models are supported with a material properties database. As we discussed above, the 3-D CAD models define both the target geometry and the materials assignments. We chose our modeling paradigm to be compatible with the AFRL Satellite Assessment Center (SatAC) modeling work for several reasons. First, SatAC offers considerable existing modeling expertise, materials database resources, capabilities, and experience; it currently builds the only nationally validated satellite CAD models accepted by the National Air Intelligence Center (NAIC). When LRST is used operationally, it too will need to import only those target models that have been validated, blessed, and configuration managed by NAIC. Second, SatAC does considerable high energy laser (HEL) weapon vulnerability modeling using these same CAD models. Finally, this route will ultimately allow LRST to be applied to other HEL programs, such as the Space-Based Laser (SBL) or Tactical High Energy Laser (THEL) with the confidence that the target models will still be immediately compatible.

We will start with a general discussion of the BRDF and the usual approximations one makes. We will not derive the MB polarized BRDF equations in detail from first principles, nor will we discuss extracting model parameters from measured laboratory data, because these were described in detail in [3]. For LRST type applications, the wavelength is very narrow band, and is centered at the laser wavelength. In fact, all target materials should be directly measured at all the relevant Airborne Laser (ABL) system wavelengths. However, other signature application codes need to model the BRDFs at many other wavelength bands, whether the BRDF data are measured or not. Therefore, we need to discuss how to interpolate/extrapolate the BRDFs to other unmeasured wavelengths. In LRST we fundamentally classify the scattered energy into two regimes, diffuse and specular. Diffuse goes into a full hemisphere, but specular goes predominantly into a restricted forward cone along the nominal specular direction. We will present algorithms that determine this effective specular lobe cone angle.

In the best of all situations we would have complete measurements for all possible combinations of wavelength, polarization, and incident and reflection geometry angles to accurately model the radiometric scatter for any material type and shape. Unfortunately this clearly is not possible due to the large number of variables, the relative difficulty and high cost of making precise BRDF measurements, and the unmanageable amount of digital data that would result. The BRDF is fundamentally driven by the physical state of random (isotropic or otherwise) surface roughness. This situation would therefore be further compounded by the infinite variety of material finishes possible for a single material type (e.g., aluminum milled, brushed, polished, anodized, or even painted). For semi-transmissive materials or coatings such as plastic or paint, more than one

BRDF is required to complete the polarized scatter model, making things even more complicated.

While the possibilities for a physical BRDF are countless, most man-made materials manifest a BRDF functional form that is relatively simple and usually well behaved. The scatter usually decomposes down into two basic forms – wide-angle depolarized diffuse, and narrow-angle polarized specular. Diffuse reflections mostly depolarize the reflected light, while specular reflections tend to preserve the incident polarization. It has been shown that the Maxwell-Beard BRDF model allows us to take a minimum number of measured material optical properties in the laboratory, and accurately predict the *polarized* scatter for any combination of source-receiver geometry, using first principles and reasonable approximations where necessary. The primary advantage of this model over other BRDF models that do limited parameter fits is its ability to usually reproduce the original measured BRDF data almost exactly, especially for the monostatic direction. Because the BRDF curves are measured, the model inherently tends to be self-correcting when the actual scattering physics deviates from the assumed underlying first principle processes within the basic equations. In other words, because the MB BRDF model uses look-up tables taken from measured data, it tends to capture scattering physics not originally contained in the starting equations. The MB BRDF formalism starts with a reasonable and rigorous physical model, and ends up being a good parameter fit model.

No BRDF discussion would be complete without the usual warnings about taking BRDF data, whether measured or simulated, too literally. BRDF data apparently are very difficult to measure accurately by anyone. Historically, round-robin measurement programs (in which a material sample is passed from laboratory to laboratory to have its BRDF measured) do not return consistent data. Between laboratories there are the usual differences between mechanical resolution, aperture size, stray light, detector noise levels, how the sample is held (stationary, spun), where on the sample the laser spot is placed, calibration specimens, optics, and polarization elements. Finally, there is an issue with the exact, current, real-world state of a target material surface. How old is the surface finish? Has it been changed by weather, oxidation, handling, sand-blasting, or heat? Just how valid are the accurately measured BRDFs that we use to do the simulation modeling? Do all targets of the same class really share the exact same BRDFs?

Now, this is not to say that we know little about the target BRDFs. In general, a material's reflective property tends to become more diffuse with time. For example, it is difficult to maintain a clean shiny surface on a missile, especially if the missile is kept outside subject to weather and oxidation. Therefore, if we use measured BRDF data from pristine material coupons, we are likely to overestimate the specular and underestimate the diffuse contributions. For the purposes of a safety analysis, this is probably adequate. We will make predictions closer to a hazard level in the specular directions, where it is the most important to be conservative, and slightly under-predict hazards in the less critical diffuse directions. We must be prepared for possible errors in the radiometry on the order of factors of 2 or more, due to our lack of knowledge of the exact target BRDFs.

We should always strive to use the best engineering knowledge of the scenario and BRDFs. But we also need to recognize and accept some level of uncertainty in our simulation, balancing the simulation's fidelity against what we know about the uncertainties.

The MB BRDF model relies entirely on in-plane measurements. That is, the receiver sensor measures the reflected intensities at various scan positions in the plane of incidence of the laser source. The plane of incidence contains the incident laser light rays and the material surface normal vector. Classically the source is polarized with the electric vector axis either parallel (i.e., P polarization) or perpendicular (i.e., S polarization) to the plane of incidence. The receiver can also resolve the scattered light into one of these two directions using a linear polarizer. Usually four in-plane scans are made to cover each combination: S to S (SS or $\perp\perp$), S to P (SP or $\perp\|$), P to P (PP or $\|$), and P to S (PS or $\|\perp$). From these measured data, MB model parameters are extracted and put into a material properties database. Simulation programs can then access the material properties, on a material code number basis, and both reconstruct the BRDF for in-plane geometries, and predict (i.e., extend) the BRDF for out-of-plane and bi-static geometries.

Going back several hundred years, it was recognized that the surfaces of opaque materials reflected visible light from a large number of microscopic facets (micro-facets) inherent in randomly rough surfaces. The dominant effect is for light to specularly reflect off these little mirror facets (or sequins) in all directions, the angular spread being driven by the micro-facet tilt statistics. Since the micro-facets are very smooth at the microscopic level, the polarization then tends to follow the Fresnel laws of reflection.

In fact, the primary assumption in the Maxwell-Beard BRDF model is that within the visible and infrared wavelength regions, the micro-facet reflection process is entirely specular, geometric (i.e., minimal wave effects), and driven by Fresnel's laws at the first air-material surface interface. Light is scattered out-of-plane by micro-facets tilted out of the plane of incidence, for instance. Figure 4.7 depicts how only a relatively few (correctly tilted) micro-facets actually reflect light back to a sensor.

The net scatter from dielectric materials then includes two reflections, one due to the first-surface Fresnel reflection, and a second due to multiple reflections off pigment particles in the shallow volumetric region just below the first-surface. Figure 4.8 gives a notional depiction of how just one ray of light might scatter back to the sensor.
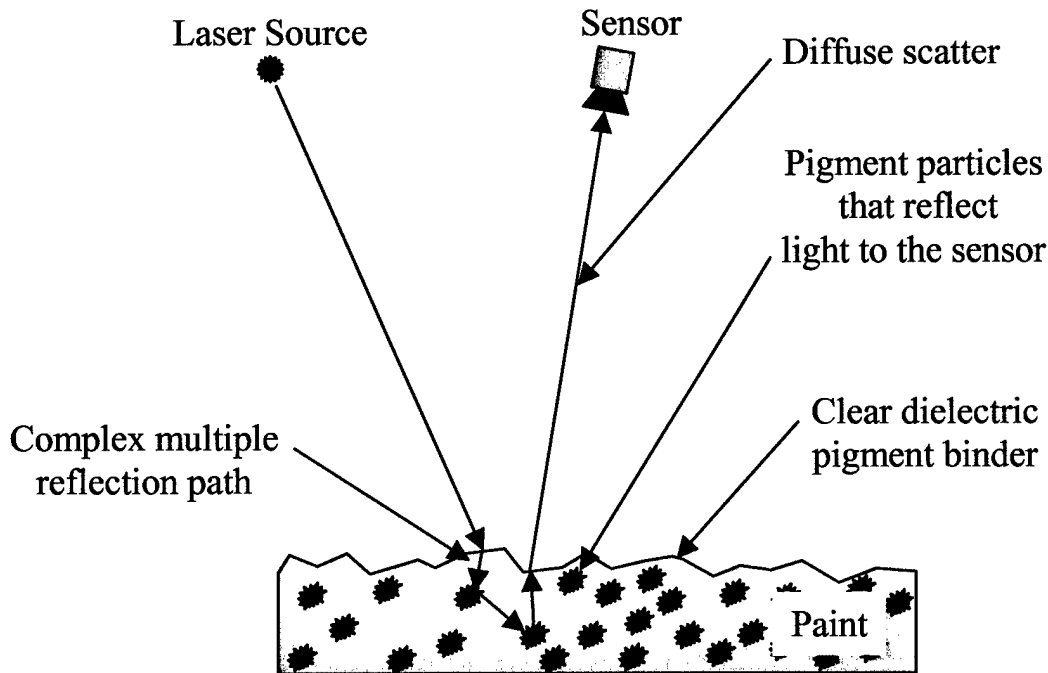
Figure 4.8. Volumetric Scatter from Pigment Particles

The first-surface reflection, therefore, is again a function of the distribution of randomly tilted, specularly reflecting, dielectric micro-facets. Given the source and receiver directions, only those micro-facets whose local surface micro-facet normal bisects the incident and reflected directions contribute to the instantaneous scatter measured at the receiver aperture. Thus, the scatter intensity is proportional to the number of correctly tilted micro-facets and the relative area they subtend.

Fresnel's equations are used to handle the polarization effects. The first-surface scatter is assumed to be always polarized because Fresnel's equations do not provide for depolarization. When the first-surface becomes very rough (e.g., has deep grooves), light incident from directions near normal tend to bounce more than once from the highly tilted micro-facets. This both depolarizes and diffusely distributes the emerging light, which contradicts the above assumption of single reflections off micro-facets that preserve polarization. Shifting this fraction of the scatter into the depolarized volumetric diffuse terms captures this effect.

Contrary to MB assumptions, in reality wave effects are always present. A certain amount of diffraction always exists and tends to spread the specular lobe slightly. While the purely geometric assumption is violated (albeit only slightly for most materials), the model inherently
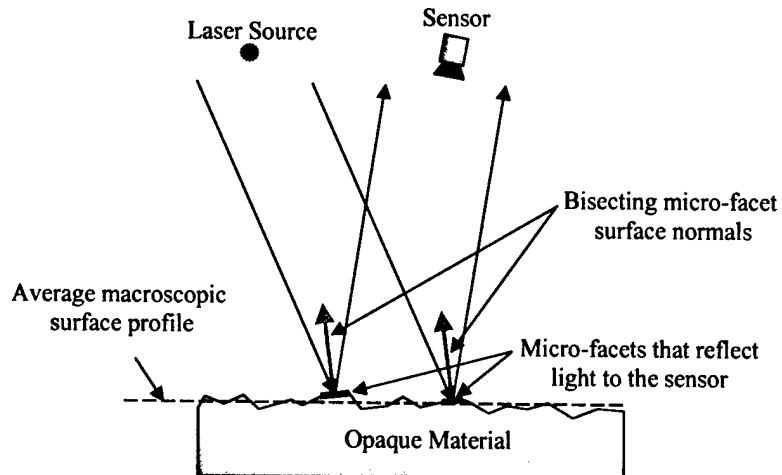
Figure 4.7. Basic first Surface Scatter. Only a Relatively Few Micro-facets Bisect the Directions from Source to Receiver

These few physical assumptions seem to be well satisfied by most man-made machined materials with a fairly smooth finish. However, extremely rough surface finishes lead to complex multiple reflections of the light within the first-surface micro-structure, and hence both depolarize the light and spread the distribution of scattered light into all directions more uniformly (i.e., diffuse scatter). Extremely rough surfaces are not well modeled by the Maxwell-Beard first-surface BRDF model except where the derived parameters are especially crafted to follow a diffuse distribution. At the other extreme, when the first-surface becomes very smooth compared to the wavelength, wave effects begin to become more important than the geometric effects. Also, the MB BRDF model cannot easily handle mirror-like surface finishes (roughness very much less than the wavelength). Only optical elements such as mirrors and lenses have both smooth surface finishes and well-controlled surface figures. Therefore, it is not likely that an exterior material for a missile will have a material finish that is optically shiny and which supports coherent far-field constructive interference.

If the material is a metal, then light penetrates the first-surface to a depth on the order of only a few wavelengths. The entire scattering process is contained in just the single interaction of light with the opaque first-surface micro-facets. For paints, the first-surface reflects a small amount, and transmits most of the incident light into its bulk region.

Dielectric materials such as paints and plastics therefore exhibit a more complex reflection mechanism. On a microscopic level, light can penetrate and undergo very complex multiple reflections off pigment particles suspended in the clear dielectric plastic binder within a shallow volumetric region behind the first-surface. The detailed interaction process would be extremely difficult to model in detail, even with very comprehensive measured material property information (which of course *never* seems to be available). The MB BRDF model approximately captures this complex volumetric scatter physics by using two diffuse BRDF functional forms whose parameters are derived again from the in-plane scan data.

recovers somewhat by relying on the actual measured scan curves, wave effects measured and therefore included.

We can start our mathematical discussion by stating a useful definition of the BRDF when the incident light is collimated:

$$\text{BRDF} = \rho'(\theta_i,\ \varphi_i,\ \theta_r,\ \varphi_r) = \frac{\delta L_r}{\delta E_i} = \frac{\dfrac{\delta P_r}{\delta A \cdot \cos(\theta_r) \cdot \delta\Omega_{rec}}}{\delta P_i / \delta A} \qquad (\text{sr}^{-1}) \qquad (4.25)$$

where,

δL_r	= the incremental reflected radiance from a flat surface (W/m^2-sr)
δE_i	= the incremental incident irradiance (W/m^2)
δP_i	= the power incident on the differential area δA
δP_r	= the power reflected into the incremental solid angle $\delta\Omega_{rec}$ from area δA (W)
θ_i	is the polar angle of the incident light (down from the local surface normal)
φ_i	= the azimuthal angle of the incident light (measured from some local in-plane axis)
θ_r	= the polar angle of the receiving aperture
φ_r	= the azimuthal angle of the receiving aperture
δA	= the differential area of the reflecting macroscopic surface (m^2)
$\delta\Omega_{rec}$	= the differential solid angle subtended by the receiving aperture (sr)

We will freely interchange our notation of the BRDF with ρ'. The BRDF stated in Equation 4.25 is simply a ratio of reflected directional radiance to incident irradiance. We will want to add to our notion of the BRDF by keeping track of variables such as polarization, wavelength, material type, specular and diffuse contributions, and so forth. Notice that any dependence on the incident cone solid angle has been simplified to $\delta P_i/\delta A$. For most laser illumination scenarios, certainly for the ABL ones, the incident light is collimated to within tens of micro-radians across the whole of the target, and extremely collimated locally. The effective BRDF for light sources such as the sun with a finite extent (i.e., a finite incident solid angle) can be constructed from the collimated BRDF if necessary.

A real BRDF is typically a function of a large number of variables, too large to manage without making some assumptions. Geometric variables include the azimuth and polar angles of the incident light; the azimuth and polar angles of the reflected direction; the local surface curvature; and the specific location on a surface element. Other important variables include wavelength band (which leads to an effective reflectivity), complex indices of refraction, incident polarization state, receiver polarization sensitivity, material type or detailed microstructure, material finish state (e.g., smooth or rough), and shadow/masking effects. We typically have the capability to generate separate BRDF objects for different materials, at different wavelengths; however, directional or polarization effects are derived from the basic measured BRDF data. To

keep the amount of required supporting measured data manageable, we will need to make some assumptions and reasonable approximations.

We will simplify our radiometric equations by first assuming that the geometry follows reasonably closely that encountered for the ABL scenarios of interest. We will always assume that the incident rays are parallel, with the same polarization state. This is the far-field assumption for the laser source. The difference in the incidence angles for rays at the nose or tail of a missile is tens of micro-radians (at ABL ranges of 100 km or more), which is well below the angular spread of even a very shiny surface (e.g., specular lobe width of 0.1 degree). Likewise, we assume that the receiving aperture is at a sufficiently large distance from the scattering surfaces that all the reflected rays are essentially collimated at the receiving aperture. This has a twofold importance in our equations. First, because the receiving aperture is again assumed in the far field of the scattering target, each unit vector along the reflected direction from each small scattering incremental area on the target to the receiving aperture is parallel to the other unit vectors. Second, the dimension of the collecting aperture is assumed to be much smaller than the BRDF angular spread. This ensures that we can accurately calculate the reflected intensities at the receiver aperture using a linear approximation. We calculate the irradiance at a single point, located at the center of the receiving aperture, and linearly scale it with aperture area, without doing a detailed spatial integration over the aperture area. These two assumptions allow us to reduce the number of spatial integrals and thereby increase efficiency without losing accuracy. The motivation is to increase execution speed, and making these approximations introduces negligible error in the radiometrics (and none in the BRDF).

Next we need to assume that the BRDF is isotropic and invariant in several ways. We assume the BRDF is identical for the same material assignments for all target models. All assignments of *0001 Aluminum Alloy, 2024-T3, Polished*, in the materials database for example, should render the same because the BRDF data are the same. This assumption breaks down when one considers the life history of a particular target in the real world unless it is somehow sealed off from the environment. We assume that the BRDF is invariant over the entire surface of the element it is assigned to. Variations in surface finish or color are assumed small for even large area coverage. We must assume that the micro-facet tilt distribution function is statistically isotropic in azimuth. This removes any absolute azimuthal dependency of the BRDF. Some metal finishes, for example, tend to have striations that are machine-tool grooves. This tends to produce anisotropic BRDFs that vary with the azimuth angle that is tied to surface coordinates. The BRDFs we measure are all taken at room temperature and represent how the surface scatters light at normal temperatures. At extreme melting temperatures the BRDF is clearly going to make nonlinear changes. Usually, but not certainly, the color will turn darker and the surface will become rougher because of the degradations cause by elevated temperature. For the current version of LRST, we assume that the BRDFs are invariant with temperature. Each of these assumptions is fairly reasonable for most cases, though there are exceptions. Common techniques–such as texture mapping–do exist for handling local variations, but LRST, for now, will adhere to these isotropic assumptions and not implement more complex algorithms.

A real BRDF is usually highly complex. The possibilities for scatter are literally endless. However, we need to reduce the complexity to something manageable in code. We assume that all BRDFs are well behaved and have a few common properties. We always use measured data

where it exists. There is always a certain amount of data smoothing and averaging when extracting representative optical parameters. We always assume that the BRDF is separable into two components: a polarized first-surface specular lobe that scatters light into a cone about the nominal specular direction; and a depolarized hemispheric diffuse scatter that basically follows a Lambertian distribution, coming from the volumetric region (or a very rough first surface). We are basically considering just two common material types, metals and opaque paints or plastics. The out-of-plane scatter is either Lambertian, or derives from those micro-facets that are tilted out-of-plane. In either case, all scatter is mostly a direct consequence of geometric reflection and not wave effects.

We then assume that the BRDF will have approximately the same spatial distribution regardless of wavelength, as long as the wavelength is not too different from that used to measure the BRDF material properties. This allows us to interpolate a BRDF to a wavelength for which we have no measured data beyond the net directional hemispherical reflectivity (DHR). We shall borrow the BRDF spatial distribution function and renormalize to the new reflectivity. We also need to recognize that it is the complex indices of refraction that determine the hemispherical reflectivity of metals. The indices likewise determine the first-surface reflectivity of dielectrics, but they tend to be wavelength invariant, meaning that the first-surface reflectivity remains at a fairly constant low value for large wavelength shifts. The dominant reflection effect for paints and plastics is the bulk region scatter which is assumed Lambertian (or nearly so - see the MB model below), and is controlled by the bulk reflectivity.

The BRDF is commonly measured for nominally flat surface profiles. Surface curvature radiometric effects are accounted for independently of the BRDF, by either analytically integrating around the curved part, or by evaluating the radiometrics at many locally flat differential areas (i.e., essentially numerically integrating). There is definitely a coupling between curvature and the specular reflection: a shiny cylinder reflects light into a thin bright ring around itself, whereas, a flat plate projects a small round spot. Diffuse surfaces, on the other hand, are affected less by local curvature because the fall-off occurs as only a cosine of the polar angle. We will still track the diffuse scatter for small locally flat areas. We will shortly show how we can completely capture curvature effects through integrations around primitive surfaces.

Finally, we need to ensure that whatever the BRDF functional form, an integral of the outgoing scattered intensities over the hemisphere above the scattering surface, is equal to the measured hemispherical reflectivity. The hemispherical reflectivity is itself a function of wavelength, and is independently measured in the laboratory. We will need to address the normalization procedures for each of the BRDFs below, since this is the basis for our radiometric energy balance. The hemispherical reflectivity is related to the BRDF by

$$
\text{DHR}\ (\theta_{in}, \lambda) = \frac{\sum\limits_{hemisphere} \delta P_r}{\delta P_i} = \int\limits_{hemisphere} \text{BRDF} \cdot \cos(\theta) \cdot d\Omega \tag{4.26}
$$

and can be expressed more completely as:

$$\mathrm{DHR}\,(\theta_{in},\lambda) = \int_0^{2\pi} \int_0^{\pi/2} \mathrm{BRDF}\,(\theta_{in},\lambda,\theta,\varphi) \cdot \cos(\theta) \cdot \sin(\theta) \cdot d\theta \cdot d\varphi \qquad (4.27)$$

where we have indicated that the hemispherical reflectivity is a function of wavelength and the incident angle of the light. We use the notation $\mathrm{DHR}\,(\theta_{in},\lambda)$, which stands for directional hemispherical reflectivity. This quantity is a function of wavelength, but it also changes with the angle of incidence for all materials. We will consider $\mathrm{DHR}\,(\theta_{in} = 0,\lambda)$ for normal incidence.

Consider an illumination-reflection-sensor geometry like that shown in Figure 4.9. The laser is assumed to present a uniform monochromatic collimated irradiance over the test sample surface; the sample is nominally flat and isotropic; and the sensor is located in the far field. The sensor has a small receiving aperture that subtends a relatively small but non-zero collecting solid angle $\delta\Omega_r$.
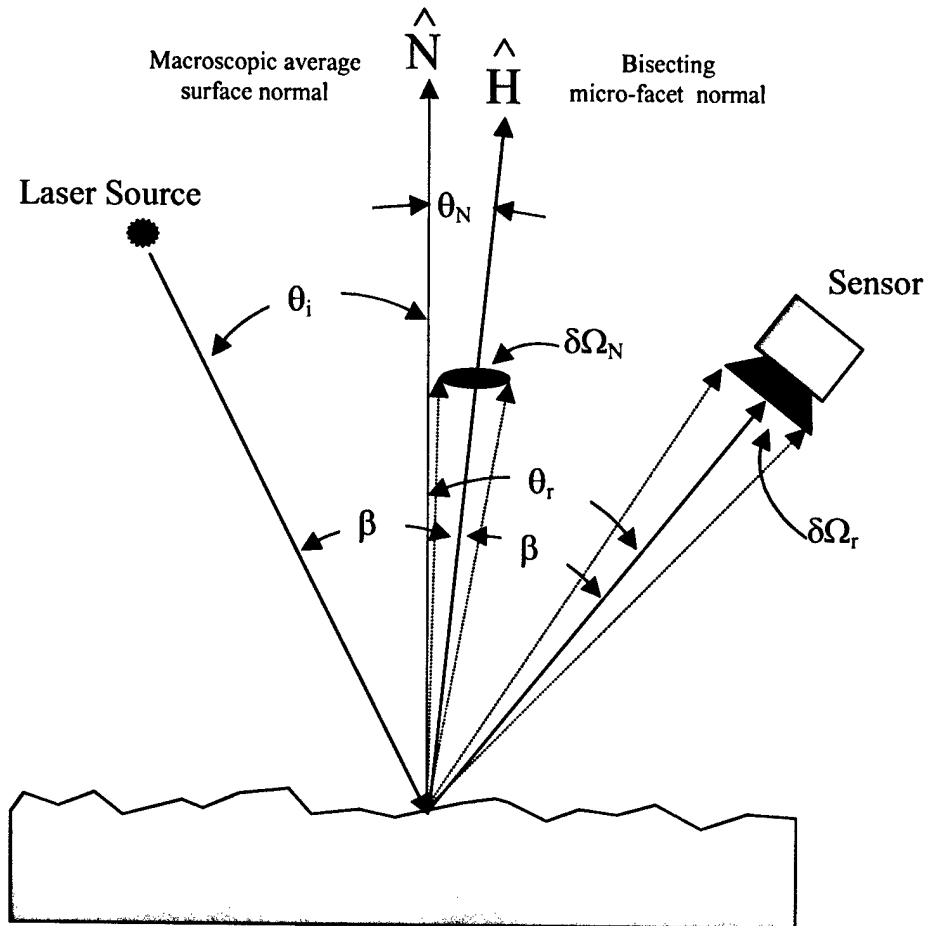
Figure 4.9. Basic Geometry and Notation for the BRDF Model Equations

Let $\Xi(\theta_N,\varphi) \cdot d\Omega_N$ represent a distribution function which gives the relative fraction of the material surface area whose micro-facets are tilted relative to the macroscopic average surface

normal, along a direction (θ_N, φ), where these are the usual polar angles, and all within a small solid-angle cone $\delta\Omega_N$. By our isotropic roughness assumption we realize in effect that $\Xi(\theta_N, \varphi) = \Xi(\theta_N)$.

All micro-facet surface normals that fall within a small solid angle $\delta\Omega_N$ centered about the bisecting direction $\hat{H}$, between light source and receiver, will scatter light (geometrically) into the receiver aperture solid angle such that:

$$\delta\Omega_r = 4 \cdot \cos(\beta) \cdot \delta\Omega_N \qquad (sr) \qquad (4.28)$$

While Equation 4.28 might not be obvious at first glance, it is a consequence of the optical gains for differential tilts of the micro-facet mirrors. We have shown [9] that a mirror that moves an angle $\delta\theta_{mirror}$ in-plane of source and receiver causes a change in the reflected ray line of sight (LOS) of $\delta\theta_{LOS} = 2 \cdot \delta\theta_{mirror}$, exactly. A mirror that moves an angle $\delta\theta_{mirror}$ cross-plane of source and receiver causes a change in the reflected ray LOS of $\delta\theta_{LOS} = 2 \cdot \cos(\beta) \cdot \delta\theta_{mirror}$, where β is the angle between the incident (or reflected) LOS and the bisecting micro-facet normal $\hat{H}$, and where small angular changes are assumed valid. A product of these two terms leads directly to Equation 4.28 above.

For a given unit surface area with light incident upon it, the micro-facets whose normals are tilted at an angle θ_N with respect to the average macroscopic surface normal will reflect light into the receiver aperture solid angle. Typically, only a small fraction of the total surface area has micro-facets that satisfy this condition. For now, assume that all such micro-facets are neither shadowed nor masked. For a small surface area element δA, the net amount of that surface area that has micro-facets tilted toward the receiver (i.e., along θ_N, δ) is defined as:

$$A(\theta_N, \varphi) = \Xi(\theta_N, \varphi) \cdot \delta A \cdot \delta\Omega_N \qquad (m^2) \qquad (4.29)$$

The net power incident on these micro-facets is simply:

$$P_{facets} = \frac{\delta P_i}{\delta A \cdot \cos(\theta_i)} \cdot A(\theta_N, \varphi) \cdot \cos(\beta) \qquad (watts) \qquad (4.30)$$

where we recognize $\dfrac{\delta P_i}{\delta A \cdot \cos(\theta_i)}$ as the irradiance perpendicular to the beam, and where $A(\theta_N, \varphi) \cdot \cos(\beta)$ is the projected area of the participating micro-facets.

We assume that the fundamental mechanism for reflection at the first air-material surface, and therefore for all polarization effects, is driven by Fresnel's equations. Consider the situation of external reflection where light is incident from the exterior of a target surface and where the incident medium is either air or vacuum (i.e., $n = 1.0$), and the second medium is either a dielectric or a metal. When the angle of incidence for light flux is θ_i, the refraction/reflection

angle is θ_r and the complex index of refraction is N, then Snell's law predicts the angle of refraction/reflection to be:

$$\frac{\sin(\theta_i)}{\sin(\theta_r)} = N = n - i\kappa \tag{4.31}$$

N is independent of the angle of incidence, and given in the usual optics form $N = n - i\kappa$, where κ is always positive. The pair of complex indices of refraction (n, κ) must reside in the first quadrant as long as the incident medium has an index of refraction lower than that of the bulk material. Thin layers are not considered here, so the index of refraction applies to the bulk of the material and does not depend on its thickness. Index pairs in the second and third quadrants (i.e., where n is negative) should not be allowed, since light would then be incident at a surface where the material is in an excited state and hence the reflectance would be greater than 1. Typically, clear dielectrics exhibit an absorption index κ equal to or nearly equal to zero. Metals, on the other hand, have a nonzero κ and are opaque (absorbing). In the infrared, some dielectrics can also become absorbing, and therefore would probably not show the usual diffuse bulk region scatter.

The reflectance functions for a smooth dielectric (or for a small differential area of dielectric on the microscopic level) with a monochromatic beam incident at an oblique angle β is designated by $R_{XX}(\beta)$, where we shall use the notation R_{SS} for the reflectivity when the incident and reflected electric field vectors are *perpendicular* to the plane of incidence, and R_{PP} for the reflectivity when the incident and reflected electric field vectors are *parallel* to the plane of incidence. The rigorous formulas for the reflected flux are:

$$R_{SS}(\beta) = \frac{(a - \cos(\beta))^2 + b^2}{(a + \cos(\beta))^2 + b^2} \tag{4.32}$$

$$R_{PP}(\beta) = R_{SS} \cdot \frac{(a - \sin(\beta) \cdot \tan(\beta))^2 + b^2}{(a + \sin(\beta) \cdot \tan(\beta))^2 + b^2} \tag{4.33}$$

where the a and b intermediate terms above are derived from:

$$x = n^2 - \kappa^2 - \sin^2(\beta) \tag{4.34}$$

$$y = \sqrt{x^2 + 4 \cdot n^2 \cdot \kappa^2} \tag{4.35}$$

$$a^2 = \frac{x + y}{2} \tag{4.36}$$

$$b^2 = \frac{y - x}{2} \tag{4.37}$$

Notice, in theory, there is no cross-coupling between the two orthogonal linear polarization states:

$$R_{SP}(\beta) = R_{PS}(\beta) = 0 \qquad (4.38)$$

The effective Fresnel reflectivity for unpolarized natural light, or circularly polarized light, is the arithmetic mean of the two reflectivities:

$$R_{unpolar} = \frac{R_{SS} + R_{PP}}{2} \qquad (4.39)$$

For angles where R_{SS} is greater than R_{PP}, unpolarized incident light gets slightly polarized.

At normal incidence, the S and P reflectivities should be the same:

$$R_{PP}(0) = R_{SS}(0) = \frac{(n-1)^2 + \kappa^2}{(n+1)^2 + \kappa^2} \qquad (4.40)$$

The normal incidence reflectivity is a function of the two indices. It is useful to solve for either n or κ, given the normal incidence reflectivity $R(0)$ and the other index. For instance, the complex index κ is:

$$\kappa = +\sqrt{\frac{R(0) \cdot (n+1)^2 - (n-1)^2}{1 - R(0)}} \qquad (4.41)$$

where we take only the positive root. Likewise we can solve for the real index n by:

$$n = \frac{1 + R(0)}{1 - R(0)} \pm \sqrt{\left(\frac{1 + R(0)}{1 - R(0)}\right)^2 - (\kappa^2 + 1)} \qquad (4.42)$$

where either root is valid, but the smaller is probably the useful one.

Upon reflection, each component of the electric field undergoes a phase retardation given by:

$$\tan(\delta\varphi_P) = \frac{2 \cdot \cos(\beta) \cdot \left[(n^2 - \kappa^2) \cdot b - 2 \cdot n \cdot \kappa \cdot a\right]}{\left[(n^2 + \kappa^2)^2 \cdot \cos^2(\beta) - (a^2 + b^2)\right]} \qquad (4.43)$$

$$\tan(\delta\varphi_S) = \frac{2 \cdot \cos(\beta) \cdot b}{\cos^2(\beta) - (a^2 + b^2)} \qquad (4.44)$$

The *relative* phase retardation between the P and S directions upon reflection is given by:

$$\Delta \varphi_{PS} = \delta \varphi_P - \delta \varphi_S = \tan^{-1} \left[\frac{\tan(\delta \varphi_P) - \tan(\delta \varphi_S)}{1 + \tan(\delta \varphi_P) \cdot \tan(\delta \varphi_S)} \right] \tag{4.45}$$

$$\Delta \varphi_{PS} = \tan^{-1} \left[\frac{2 \cdot b \cdot \sin(\beta) \cdot \tan(\beta)}{\sin^2(\beta) \cdot \tan^2(\beta) - (a^2 + b^2)} \right] \tag{4.46}$$

The net surface area that bounces light into the receiver aperture geometrically is:

$$A(\theta_N, \varphi) = \Xi(\theta_N, \varphi) \cdot \delta A \cdot \frac{\delta \Omega_r}{4 \cdot \cos(\beta)} \quad (\text{m}^2) \tag{4.47}$$

The amount of incident power at the micro-facets is:

$$\delta P_{facets} = \frac{\delta P_i}{\cos(\theta_i)} \cdot \Xi(\theta_N, \varphi) \cdot \frac{\delta \Omega_r}{4} \quad (\text{watts}) \tag{4.48}$$

The amount of power that is reflected depends on the incident polarization state. Knowing the Fresnel reflectance R_{xx} (where X can be S or P), we can determine the amount of power collected by the receiver aperture:

$$\delta P_r = \delta P_{facets} \cdot R_{XX} = \frac{\delta P_i}{\cos(\theta_i)} \cdot \Xi(\theta_N, \varphi) \cdot \frac{\delta \Omega_r}{4} \cdot R_{XX} \quad (\text{watts}) \tag{4.49}$$

Finally:

$$\rho'(\theta_i, \theta_r, \beta, \theta_N) = R_{XX}(n, \kappa, \beta) \cdot \frac{\Xi(\theta_N)}{4 \cdot \cos(\theta_i) \cdot \cos(\theta_r)} \quad (\text{sr}^{-1}) \tag{4.50}$$

where θ_i is the incident angle of the light with the average macroscopic surface normal, θ_r is the angle the receiver LOS makes with the surface normal, β is the angle between the bi-section unit vector $\hat{H}$ and either the incident or reflected LOS unit vectors, and θ_N is the angle $\hat{H}$ makes with the surface normal. The bi-section unit vector $\hat{H}$ is the same as the micro-facet local normal:

$$\hat{H} = \frac{\hat{V}_i + \hat{V}_r}{\left| \hat{V}_i + \hat{V}_r \right|} \tag{4.51}$$

where $\hat{V}_i$ is the unit vector LOS to the laser source, and $\hat{V}_r$ is the unit vector LOS to the receiver aperture. The angle θ_N is easily derived from the geometry:

$$\theta_N = \cos^{-1}(\hat{H} \circ \hat{N}) \tag{4.52}$$

where $\hat{N}$ is the macroscopic surface normal. The angle β is also easily derived from the geometry:

$$\beta = \cos^{-1}(\hat{H} \circ \hat{V}_i) = \cos^{-1}(\hat{H} \circ \hat{V}_r) \qquad (4.53)$$

To work the fully polarized reflection, Equation 4.50 is needed twice, once for each incident electric field component (i.e., R_{SS} and R_{PP}). For the LRST application, we assume for now that the incident laser beam polarization state is circularly or randomly polarized. The first-surface Fresnel reflectance then would be the mean of the S and P, and hence

$$R_{XX} = R_{unpolar} = \frac{R_{SS} + R_{PP}}{2} \; .$$

Equation 4.50 seems easy enough to use in a rendering algorithm. Only three quantities need to be measured: the two indices of refraction (n, κ), and the tilt distribution function $\Xi(\theta_N)$. In the laboratory we can only infer $\Xi(\theta_N)$ by making in-plane monostatic (sometimes called a zero bistatic, or quasi-monostatic) BRDF scans. In the monostatic measurement configuration, the incident and reflected angles are equal to each other and to θ_N. Equation 4.50 can then be inverted to give:

$$\Xi(\theta_N) = \rho'_{mono}(\theta_N) \cdot \frac{4 \cdot \cos^2(\theta_N)}{R(\beta = 0)} \qquad (\text{sr}^{-1}) \qquad (4.54)$$

where $\rho'_{mono}(\theta_N)$ represents the measured monostatic BRDF data. The Fresnel reflectivity in the denominator is for normal incidence, and hence is independent of polarization state. Typically in the rendering algorithm, $\Xi(\theta_N)$ is simply a look-up table of values derived from measured zero bistatic BRDF scans.

This discussion applies when the first-surface of the material is relatively smooth such that only a single light ray bounce is valid (which is usually the case for many materials). If, however, the first-surface is very rough such that multiple bounces occur, or the material is a semi-transparent dielectric paint, then a depolarized reflection process occurs too. In this case Equation 4.54 does not correctly give the tilt distribution function $\Xi(\theta_N)$.

Notice that $\Xi(\theta_N)$ is a purely *geometric* tilt distribution function. For surfaces that are reasonably smooth, we should be able to integrate over all the micro-facet areas, as projected down to the flat material surface plane, and get unity:

$$\iint\limits_{hemisphere} \Xi(\theta) \cdot \cos(\theta) \cdot d\Omega = 2\pi \cdot \int_0^{\pi/2} \Xi(\theta) \cdot \cos(\theta) \cdot \sin(\theta) \cdot d\theta \approx 1 \qquad (4.55)$$

Of course, as the surface becomes rougher, then this normalization might begin to break down because $\Xi(\theta_N)$ can probably no longer be properly measured due to shadow/masking effects. Extremely rough surface finishes, for which the surface is not a simple set of tilted facets, should certainly not follow the normalization of Equation 4.55. However, this is generally acceptable as long as the normalization is satisfied. In general, and except for extremely rough surfaces, Equation 4.55 should hold up.

We do not actually derive the tilt distribution function $\Xi(\theta_N)$ directly. Instead, because we do not explicitly know the indices of refraction, we must solve for the product of $\Xi(\theta_N)$ and the normal incidence Fresnel reflectance:

$$\Xi(\theta_N) \cdot R(0) = 4 \cdot \rho'_{mono}(\theta_N) \cdot \cos^2(\theta_N) \ (sr^{-1}) \tag{4.56}$$

If we substitute Equation 4.56 into Equation 4.55, we get:

$$R(0) \approx 2\pi \cdot \int_0^{\pi/2} 4 \cdot \rho'_{mono}(\theta_N) \cdot \cos^3(\theta) \cdot \sin(\theta) \cdot d\theta \tag{4.57}$$

Here again, this is only as good as the normalization in Equation 4.55.

Before we leave this discussion, we need to finalize our functional form for the specular BRDF. We will need to modify the form of Equation 4.50 to include the normal incidence Fresnel reflectance into the tilt distribution function as:

$$\rho' = \frac{R_{XX}(n_1, \kappa_1, \beta)}{R(n_1, \kappa_1, \beta = 0)} \cdot \left[\frac{R(n_2, \kappa_2, \beta = 0) \cdot \Xi(\theta_N)}{4} \right] \cdot \frac{1}{\cos(\theta_i) \cdot \cos(\theta_r)} \quad (sr^{-1}) \tag{4.58}$$

While one expects the normal incidence reflectance $R(0)$ to be the same in the numerator and denominator, this is not a requirement. We emphasized this by the subscripts in Equation 4.58 for the complex indices. The effective reflectivity in the BRDF is contained in the middle term, commonly called the bi_factor. In the materials database file (i.e., MATTER.DAT), the tabulated data has the keyword *BISTATIC*. It is the tabulated tilt distribution function essentially scaled by a reflectivity constant. The indices for the Fresnel reflectance in the bi_factor will never need to be directly known.

The leading Fresnel polarized reflectance function is now a *normalized* function. The important things to understand are these. Out to significant angles, there is very little polarization effect between the two reflected states. All the interesting action occurs at the larger angles. The relative depth of the Brewster dip (i.e., the degree of polarization) and the ratio of the normal incidence to the grazing incidence reflectance are the most important features. Metals should generally have minimal reflectance ratios, while dielectrics should be much greater.

Light that is diffusely scattered at either a rough first-surface or from within the volumetric region of the material must be modeled as a second and separate depolarized BRDF. Many

materials exhibit a diffuse scatter that follows the classical Lambert's law, where the apparent reflected radiance is equal in all directions, and the scatter follows a cosine law. The BRDF is therefore not a function of angle in this case:

$$\rho'_{Lambertian}(\lambda) = R_{diffuse}(\lambda) \cdot \frac{1}{\pi} = R_x(\lambda) \qquad (sr^{-1}) \qquad (4.59)$$

where $R_{diffuse}$ is some effective diffuse reflectivity for the material. The inverse π comes from the normalization given in Equation 4.27:

$$R_{diffuse}(\lambda) = 2\pi \cdot \int_0^{\pi/2} R_x(\lambda) \cdot \cos(\theta) \cdot \sin(\theta) \cdot d\theta = \pi \cdot R_x(\lambda) \qquad (4.60)$$

A Lambertian diffuse scatter can come from either the first-surface (i.e., a very rough finish), or the volumetric pigment particles in paint. While $R_{diffuse}$ is probably physically intuitive, we will refer to the Lambertian diffuse reflectivity as R_x, since this is a parameter found in the material properties database files.

Sometimes the diffuse scatter displays a slight non-Lambertian variation with the angle of incidence or observation. This arises from different rays penetrating more or less into the pigmented volumetric region of a paint (or plastic).

The common notation for the angularly dependent diffuse BRDF is:

$$\rho'_{non-lambertian}(\theta_i, \theta_r) = 2\frac{R_{hov}}{\cos(\theta_i) + \cos(\theta_r)} \qquad (sr^{-1}) \qquad (4.61)$$

where a single constant R_{hov} (sometimes notated R_v) is used to scale the strength of this diffuse functional term. The coefficient R_{hov} is fit to measured BRDF data. Even for metals, where there is no volumetric region to justify this functional form, we will use a non-Lambertian diffuse function in addition to a Lambertian form. The total diffuse depolarized BRDF scatter is given by:

$$\rho'(\theta_i, \theta_r, \beta, \theta_N)_{diffuse} = R_x + 2\frac{R_{hov}}{\cos(\theta_i) + \cos(\theta_r)} \qquad (sr^{-1}) \qquad (4.62)$$

A factor of 1/2 will be needed to account for only half the power going into one linear polarization state, if the receiver has such a polarizer. There are other effects that have been absorbed into these terms. For instance, the fact that a typical air-dielectric Fresnel interface reflects about 4% has implicitly been absorbed into the R_x and R_{hov} coefficients.

At this point we need to summarize the BRDF equations in their final form. We took time to develop the underlying physics and mathematics of the equations to help the user better understand how the functional forms arise. The net BRDF will always be a simple sum of the

polarized specular and depolarized diffuse BRDF terms. The reflected intensities depend on the incident polarization state and the receiver polarization sensitivity. The six cases are:

Incident P:Receiver P

$$\rho = \frac{R_{PP}(\beta)}{R(0)} \cdot \frac{bi\_factor(\theta_N)}{\cos(\theta_i) \cdot \cos(\theta_r)} \cdot S_o(\theta_N, \beta) + \frac{1}{2}\left[R_x + 2\frac{R_{hov}}{\cos(\theta_i) + \cos(\theta_r)} \right] \quad (\text{sr}^{-1}) \quad (4.63)$$

Incident P:Receiver S

$$\rho' = \frac{1}{2}\left[R_x + 2\frac{R_{hov}}{\cos(\theta_i) + \cos(\theta_r)} \right] \quad (\text{sr}^{-1}) \quad (4.64)$$

Incident P:Receiver none

$$\rho' = \frac{R_{PP}(\beta)}{R(0)} \cdot \frac{bi\_factor(\theta_N)}{\cos(\theta_i) \cdot \cos(\theta_r)} \cdot S_o(\theta_N, \beta) + \left[R_x + 2\frac{R_{hov}}{\cos(\theta_i) + \cos(\theta_r)} \right] \quad (\text{sr}^{-1}) \quad (4.65)$$

Incident S:Receiver S

$$\rho' = \frac{R_{SS}(\beta)}{R(0)} \cdot \frac{bi\_factor(\theta_N)}{\cos(\theta_i) \cdot \cos(\theta_r)} \cdot S_o(\theta_N, \beta) + \frac{1}{2}\left[R_x + 2\frac{R_{hov}}{\cos(\theta_i) + \cos(\theta_r)} \right] \quad (\text{sr}^{-1}) \quad (4.66)$$

Incident S:Receiver P

$$\rho' = \frac{1}{2}\left[R_x + 2\frac{R_{hov}}{\cos(\theta_i) + \cos(\theta_r)} \right] \quad (\text{sr}^{-1}) \quad (4.67)$$

Incident S:Receiver none

$$\rho' = \frac{R_{SS}(\beta)}{R(0)} \cdot \frac{bi\_factor(\theta_N)}{\cos(\theta_i) \cdot \cos(\theta_r)} \cdot S_o(\theta_N, \beta) + \left[R_x + 2\frac{R_{hov}}{\cos(\theta_i) + \cos(\theta_r)} \right] \quad (\text{sr}^{-1}) \quad (4.68)$$

and when the incident polarization is either unpolarized or circular, and receiver none:

$$\rho' = \frac{R_{un}(\beta)}{R(0)} \cdot \frac{bi\_factor(\theta_N)}{\cos(\theta_i) \cdot \cos(\theta_r)} \cdot S_o(\theta_N, \beta) + \left[R_x + 2 \frac{R_{hov}}{\cos(\theta_i) + \cos(\theta_r)} \right] \quad (sr^{-1}) \quad (4.69)$$

where the unpolarized mean Fresnel reflectance is used in the specular BRDF.

In the LRST application, the receiver will most likely have no polarization selectivity. Whether the receiver is a biological eye or an electro-optical imaging sensor, both polarization components will be received. The incident polarization state will more than likely be elliptically polarized. For reflection geometries where the included angle between source and observer is less than 45 degrees, the polarization effects are minimal. For the more grazing geometries, the S reflectance will be higher than the P, and the partition of energy into each of these directions might be more important to consider. At the extreme grazing geometries, both reflectances go up dramatically, and again the polarization preference is less critical.

The issue then is whether to worry about polarization effects for those reflection angles for which the degree of polarization is greatest. If the laser polarization is either depolarized or circularly polarized, then using the mean unpolarized Fresnel reflectance is accurate. If the laser polarization is elliptical, then using the mean unpolarized Fresnel reflectance is probably still adequate. If the laser polarization is strictly linear, then there could potentially be a some reflectance difference for some materials; the error occurs for mid-angles (e.g., 45–85 degrees), and is the difference between the unpolarized mean Fresnel curve and the S and P curves. For a linear polarized axis at 45 degrees to the surface normal, the equations are implemented in the code will predict values that are low for the S reflection, and high for the P reflection. The materials that have significant first-surface specular reflections are metals. The interesting thing is that most reflective metals have Fresnel curves that do not separate very much because the S and P reflectances are already fairly high. The approximation in using the unpolarized mean Fresnel reflectance for all cases should be a good compromise.

There are three approaches to capturing the polarization effects. First, we could model the polarization effects in detail. This would involve accurately modeling the incident polarization state of the laser (which would be determined by ABL performance tests, for example). Either a Stokes vector or a coherency matrix would represent this state. Next, at each reflection location on the target surface, we would need to rotate the frame of the incident polarization state vector into the local frame of the S and P directions. This then gives us the partition of power in the two orthogonal polarization states *(S* and *P)* such that we can separately weight the respective polarized BRDF. The downside to this approach is the rather expensive computational cost of always solving for the S and P weighting.

The second approach involves no additional computation, and is really the most conservative. We can always use just the S reflectance curve and accept the error for always overestimating the P reflectance.

The third approach also involves virtually no additional computational cost and should give higher-fidelity results. Use the unpolarized or mean Fresnel reflectance value instead of the S

reflectance. On average this should balance the errors in overestimating the P and under-estimating the S reflectances. This is the approach currently used in LRST.

In addition to the ERIM BRDF model, we currently use two other simplified models that are procedural and require only two parameters to completely specify the bi-static BRDF functional form. Having a suitable zero-order BRDF model is important for several reasons. We are often faced with having no measured BRDF data for some materials; this is especially true for foreign technology items. Sometimes we want to do parametric studies to assess the effect of a BRDF on a particular signature. It is useful to have a BRDF functional form that lends itself to analytic manipulation.

The first model is a modified Phong BRDF model [6] commonly found in the computer graphics literature. Phong basically models the specular effects by a simple cosine raised to some power, which is computationally cheap because the cosine is readily available from routine dot products. The modification we made was to bend this model to look very much like the Maxwell-Beard model, in which the cosine factor substitutes for the bi_factor. We ensure that the BRDF complies with the standard hemispherical reflectivity normalization requirement.

The Phong model is a simple two-parameter empirical model that attempts to describe the BRDF as having a diffuse (i.e., an asymptotic Lambertian) component, and a shiny component around the specular reflection direction.

The specular contribution to the BRDF is responsible for giving smooth surfaces their shiny look. The primary difference between BRDF models usually centers on the detailed functional form for the specular component. The Phong model attempts to model the specular fall-off by a cosine raised to some power. The larger the exponent, the more specular the BRDF appears.

The Phong formulation starts by constructing a bisection vector $\hat{H}$ between the incident light direction and the observer direction; then the angle between $\hat{H}$ and the surface normal is θ_N, the angle used as the argument of the cosine. The Phong BRDF assumes that the bi_factor (i.e., the tilt distribution function) is this cosine power function:

$$\rho'_{XX}(\theta_i, \theta_r, \beta, \theta_N)_{specular} = \frac{R_{XX}(n, \kappa, \beta)}{R(0)} \cdot M \cdot \cos^n(\theta_N) \cdot \frac{1}{\cos(\theta_i) \cdot \cos(\theta_r)} \cdot S_o \qquad (\text{sr}^{-1})$$

$$(4.70)$$

where M is a normalization constant (it conserves the total specular reflectivity), and $M \cdot \cos^n(\theta_N)$ is the effective tilt distribution function. The leading function is the normalized Fresnel reflectivity. The trailing function S_o is the shadow/masking term that is everywhere defaulted to 1. We do not implement S_o because the change in results by including it would be well within the current error range of the bi_factor.

For small values of n, the specular nature of the BRDF is not very pronounced. Figure 4.10 plots a few examples of the specular BRDF function for varying values of n when the light is incident vertically.
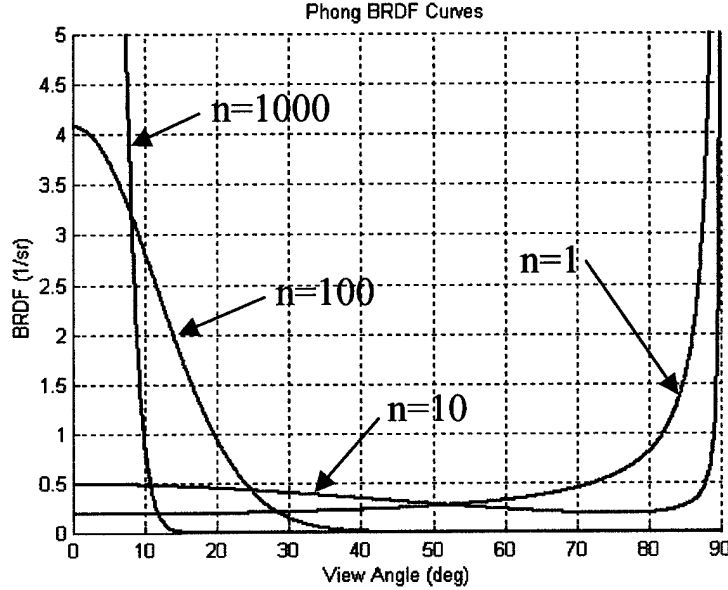
Figure 4.10. Phong Specular BRDF for: n=1000,100,10, and 1, Assuming Unity Reflectivity

The up-turn of the BRDF at large angles is typical of BRDF models, and results from the $\cos(\theta_r)$ in the denominator. This is especially true when the bi_factor predominantly scatters into large angles (i.e., small values of n). The physical quantity of interest, of course, is "intensity," not the BRDF. The BRDF turn-up is always balanced by a cosine in the numerator that prevents unbounded intensities at grazing angles.

As an approximation, the specular value drops to half the peak when the angle is

$$\theta_{HWHM} = 2 \cdot \cos^{-1}\left[10^{\frac{-0.30103}{n}} \right] \qquad \text{(rad)} \qquad (4.71)$$

For example, let us take $n = 100$. The half intensity – half angular width would be 13.47 degrees, which agrees with Figure 4.10.

The Phong model also supports a diffuse BRDF component, which is *always* Lambertian:

$$\rho'_{Lambertian} = R_x \qquad \qquad \text{(sr}^{-1}\text{)} \qquad (4.72)$$

The diffuse BRDF parameter R_x is not directly supplied by the materials database - rather, only the relative fraction of first-surface diffuse scatter is specified. The volumetric diffuse (if the material is a paint) is inferred during the normalization.

The Gaussian BRDF is probably the most suitable, best-behaved functional form we have for comparing numerical predictions with analytic formulas. It normalizes very well at all angles of incidence (as long as the surface is relatively shiny). However, notwithstanding its convenient analytical properties, we should expect only qualitative agreement of this model with nature.

The Gaussian formulation also starts by constructing a bisection vector $\hat{H}$ between the incident light direction and the observer direction. The angle between $\hat{H}$ and the surface normal is θ_N, the angle used in the exponential argument. The specular BRDF is now given by:

$$\rho'_{XX}(\theta_i, \theta_r, \beta, \theta_N,)_{specular} = \frac{R_{XX}(n, \kappa, \beta)}{R(0)} \cdot \frac{M \cdot e^{-2 \cdot \theta_N^2 / \sigma^2}}{2 \cdot \pi \cdot \sigma^2} \cdot \frac{1}{\cos(\theta_i) \cdot \cos(\theta_r)} \cdot S_o \qquad (sr^{-1}) \quad (4.73)$$

where M is a normalization constant, and $\dfrac{M \cdot e^{-2 \cdot \theta_N^2 / \sigma^2}}{2 \cdot \pi \cdot \sigma^2}$ is the effective tilt distribution function. The leading function is the normalized Fresnel reflectivity. The trailing function S_o is the shadow/masking term that is also defaulted to 1 everywhere. The argument to the exponential contains σ, which controls the angular width of the specular lobe, and is specified in units of radians.

Figure 4.11 shows examples of the Gaussian BRDF shape. The BRDF falls to $e^{-2} = 13.5\%$ of the peak when $2\sigma = 2\theta_N$. For example, the second curve with $\sigma = 10°$ reaches the e^{-2} point at exactly 20 degrees.
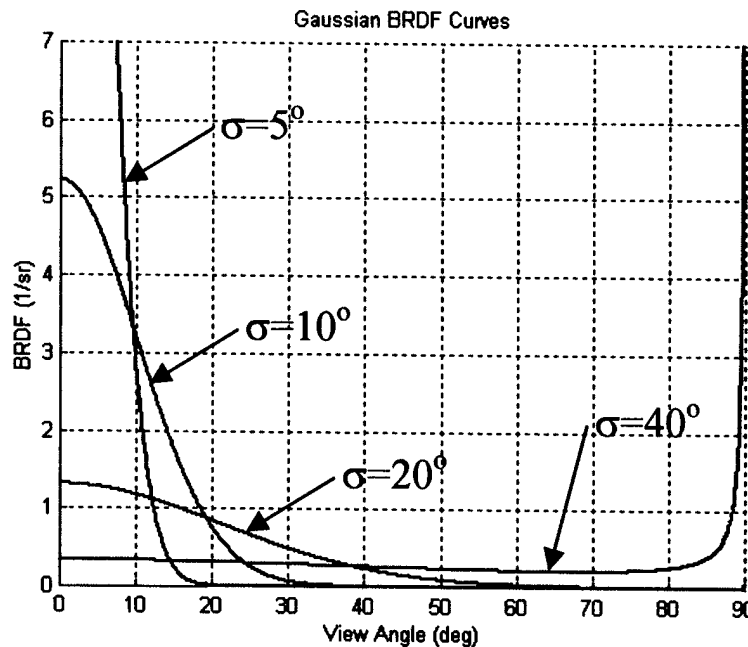
Figure 4.11. Gaussian BRDF Curves for 5-, 10-, 20-, and 40-degree σ Parameters

The Gaussian model must support a diffuse BRDF component, which is again *always* Lambertian:

$$\rho'_{Lambertian} = R_x \qquad (\text{sr}^{-1}) \qquad (4.74)$$

The diffuse BRDF parameter R_x is not directly supplied by the materials database; rather, only the relative fraction of first-surface diffuse scatter is specified. The volumetric diffuse parameter (if the material is a paint) is inferred during the normalization.

The Maxwell-Beard BRDF is typically measured at only a few wavelengths because of cost or equipment considerations. The radiometric simulation codes, however, need to make radiometric predictions for any wavelength within the stated limits of the simulation application, which usually ranges from the UV to the LWIR. We will therefore need a methodology for synthesizing a reasonable approximation to each material BRDF, as it is called out in the target model, at wavelengths where data was not directly measured. This will require making some physical assumptions and relying on the fact that the optical properties of most common materials behave reasonably well and that color and geometric spread are usually decoupled.

While we cannot always measure BRDFs at all wavelengths, we can easily measure the hemispherical reflectivity for each material at closely separated wavelengths over a very wide spectral bandwidth. This color curve is used to tie to reality the BRDF we synthesize at any wavelength within this band. Whatever BRDF interpolation scheme we decide on will certainly introduce some deviation between model and reality. At a minimum we can at least guarantee that the energy balance is preserved. In fact, even at the measurement wavelengths for the BRDFs, we will still renormalize the BRDF to this color curve for consistency.

The synthesis process is done in two steps. First we derive BRDF model parameters through interpolations; second, we renormalize the BRDF to the color curve. In some cases, the interpolation step is quite similar to the renormalization. When we have BRDF measurements that bracket the desired wavelength, we will *interpolate* the respective BRDF parameters and *renormalize* the BRDF to match the measured total hemispherical reflectivity. If the desired wavelength is outside the measured BRDF set, we will use the nearest single BRDF measurement and just renormalize the hemispherical reflectivity. We will not extrapolate BRDF parameters.

The Maxwell-Beard BRDF model is based on the theoretical assumption that the first-surface scatter is driven primarily by a purely geometric micro-facet specular-ray reflection process, and the polarization is driven by the normalized Fresnel equations for these facets. This assumption works well when the RMS surface roughness is on the order of a wavelength or larger, where diffraction effects are less important. For most target object materials the surface roughness is usually much greater than $1\,\mu m$ RMS, which means the model will work well out into the MWIR and perhaps at longer wavelengths. Surfaces smoother than the wavelength are locally approaching optical-quality polish, diffraction does become important, and the BRDF specular spread does tend to couple with wavelength. Glass would be a common example for which the typical surface roughness is much less than the wavelength, even in the visible region. Many

materials appear quite smooth in the LWIR region because of the longer wavelength, while having a broad BRDF in the visible.

The Maxwell-Beard BRDF model puts its wavelength-dependent reflectance into the bi_factor, which is the product of the tilt distribution function and the normal incidence reflectance. The bi_factor measured for short wavelengths (e.g., near-IR or visible wavelengths) has basically the same shape but is scaled linearly with changing reflectivity, by wavelength. At the longer wavelengths where diffraction and wave effects become important, the bi_factor no longer represents purely geometrical scatter by tilted micro-facets. Here the bi_factor begins to represent both an *effective* facet distribution and a reflectance. This is why a single BRDF measurement at a single wavelength is not sufficient if one expects to model radiometric quantities at both the shorter and the longer wavelengths.

The approach taken to interpolate the BRDF at a wavelength where measured data does not exist, is to synthesize a single new material, and then renormalize it to the respective DHR. Our desire is to avoid any discontinuous behavior, and to preserve the underlying detailed physical changes occurring as the wavelength changes between the bracketing measured BRDF wavelengths. Each parameter will be interpolated to avoid the discontinuous jumps. We will also need to pay attention to the material type to differentiate how we renormalize and balance the effects.

For the Maxwell-Beard BRDF, we start the parameter interpolations for the bi_factor look-up table. The first assumption is that the tilt distribution function changes smoothly and linearly from one wavelength to the next. For the shorter wavelengths, it should change very little. The bi_factor, however, includes the normal incidence Fresnel reflectance, which is also changing with wavelength. So, we first try to estimate $R(0)$ by integrating the bi_factor:

$$4 \cdot \int bi\_factor(\theta_N) \cdot \cos(\theta) \cdot d\Omega = R(0) \cdot \int \Xi(\theta_N) \cdot \cos(\theta) \cdot d\Omega \approx R(0) \quad (sr^{-1}) \quad (4.75)$$

and then minimize its influence by weighting the interpolation factors as:

$$\Xi(\theta_N, \lambda_{desired}) = \frac{(1.0 - Q)}{R_1(0)} \cdot bi\_factor(\theta_N, \lambda_1) + \frac{Q}{R_2(0)} \cdot bi\_factor(\theta_N, \lambda_2) \quad (4.76)$$

The interpolated $R(0)$ is:

$$R(0, \lambda_{desired}) = (1.0 - Q) \cdot R_1(0) + Q \cdot R_2(0) \quad (4.77)$$

Finally, the new interpolated bi_factor is the product of the new tilt function and the new $R(0)$:

$$bi\_factor(\theta_N, \lambda_{desired}) = R(0, \lambda_{desired}) \cdot \Xi(\theta_N, \lambda_{desired}) \quad (4.78)$$

This should capture the changes in the shape of the *effective* tilt distribution function. If the material is a dielectric, then the interpolated bi_factor probably should not be altered further. The normal incidence Fresnel reflectivity is already contained in the bi_factor, and it typically changes little with wavelength. If the material is a metal, then we will rescale the bi_factor later during the renormalization as required. Does the angular spread in the BRDF vary linearly with

increasing wavelength as the interpolation implies? The answer is probably no. But this approximation is probably better than making no effort to account for the narrowing of the BRDF as the wavelength gets longer. If more accuracy is required, then more BRDF data need to be taken at intervening wavelengths.

The diffuse parameters R_x and R_{hov} change with wavelength differently than the specular parameters. If the material is a metal, then we linearly interpolate them as we did in Equation 6.6 for $R(0)$. If the material is paint, then they probably change according to pigment interactions independent of any first-surface effects. We linearly interpolate the diffuse parameters in either case:

$$R_x(\lambda_{desired}) = (1.0 - Q) \cdot R_x(\lambda_1) + Q \cdot R_x(\lambda_2) \tag{4.79}$$

$$R_{hov}(\lambda_{desired}) = (1.0 - Q) \cdot R_{hov}(\lambda_1) + Q \cdot R_{hov}(\lambda_2) \tag{4.80}$$

These parameters were extracted as a pair using a least-squares fit, and should probably always be treated as a pair when doing any linear rescaling.

The shadow/masking parameters (τ, Ω) have a second order influence on the BRDF away from the specular direction. They are extracted using a least-squares fit and should probably be interpolated the same way the bi_factor is:

$$\tau(\lambda_{desired}) = (1.0 - Q) \cdot \tau(\lambda_1) + Q \cdot \tau(\lambda_2) \tag{4.81}$$

$$\Omega(\lambda_{desired}) = (1.0 - Q) \cdot \Omega(\lambda_1) + Q \cdot \Omega(\lambda_2) \tag{4.82}$$

The Fresnel indices of refraction should probably not be linearly interpolated independently of each other. Once again, without explicit measurements at the desired wavelength, how do we choose them to both satisfy the reflectivity dependence and the polarization effects (e.g., the Brewster angle)?

We want the polarization effects at the desired wavelength to be somewhere between those at the bracketing wavelengths. There are two options: hunt for a new index pair that generates an intermediate Fresnel curve, or simply pre-calculate a weighted normalized Fresnel curve based on the two normalized Fresnel curves at the bracketing wavelengths. The second approach is easier and more direct.

$$R_{SS}(\lambda_{desired}) = (1.0 - Q) \cdot R_{SS}(\lambda_1) + Q \cdot R_{SS}(\lambda_2) \tag{4.83}$$

$$R_{PP}(\lambda_{desired}) = (1.0 - Q) \cdot R_{PP}(\lambda_1) + Q \cdot R_{PP}(\lambda_2) \tag{4.84}$$

and we can extend this to the normalized Fresnel reflectivity:

$$F_{SS}^{normalized}(\beta) = \frac{R_{SS}(\beta, \lambda)}{R(0, \lambda)} \qquad F_{PP}^{normalized}(\beta) = \frac{R_{PP}(\beta, \lambda)}{R(0, \lambda)} \tag{4.85}$$

The resulting interpolated normalized Fresnel curve might not be physical, so we could again do a least-squares fit to find the closest physical index pair. However, this is unnecessary, since the error would be very small.

Now that we have our new estimates of the BRDF model parameter set, we can *renormalize* the BRDF model to match the measured DHR. The measured DHR is usually done with unpolarized broadband light. In principle all the scattered light is captured and measured regardless of whether it is specular or diffuse, polarized or depolarized. Since the incident light is unpolarized, we will therefore need to use the unpolarized normalized Fresnel reflectivity in the specular BRDF integral used for normalization.

The first-surface specular polarized total integrated reflectivity for normally incident light is given by:

$$R_{specular} = \int_{hemisphere} \rho'_{specular} \cdot \cos(\theta) \cdot d\Omega = \iint \frac{\rho'_{PP} + \rho'_{SS}}{2} \cdot \cos(\theta) \cdot \sin(\theta) \cdot d\theta \cdot d\phi \qquad (4.86)$$

We use the average of the two polarized BRDFs because we assume equal S and P incident light (i.e., unpolarized). We assume, of course, that the material properties are isotropic in azimuth, so the outer integral reduces to 2π. The integral becomes:

$$R_{specular} = 2\pi \cdot \int_0^{\pi/2} N_{unpolar}(\beta) \cdot \frac{bi\_factor\,(\theta_N)}{\cos(\theta_i = 0) \cdot \cos(\theta_r)} \cdot S_o(\theta_N, \beta) \cdot \cos(\theta) \cdot \sin(\theta) \cdot d\theta \qquad (4.87)$$

Clearly $\beta = \theta/2$, $\theta_N = \theta/2$, $\theta_i = 0$, and $\theta_r = \theta$. In code we integrate Equation 4.87 numerically with a fine angular step.

Next we integrate the diffuse unpolarized reflectivity:

$$R_{diffuse} = \int_{hemisphere} \rho'_{diffuse} \cdot \cos(\theta) \cdot d\Omega = \iint \left[R_x + 2 \cdot \frac{R_{hov}}{\cos(\theta_i) + \cos(\theta_r)} \right] \cdot \cos(\theta) \cdot d\Omega \qquad (4.88)$$

This integral is analytic:

$$R_{diffuse}(\theta_i) = \pi \cdot R_x + 4\pi \left[1 + \cos(\theta_i) \cdot \ln\left(\frac{\cos(\theta_i)}{1 + \cos(\theta_i)} \right) \right] \cdot R_{hov} \qquad (4.89)$$

When the laser is incident at $\theta_l = 0$, this simplifies to:

$$R_{diffuse}^{integrated}(0) = \pi \cdot R_x + 3.856026253 \cdot R_{hov} \qquad (4.90)$$

If the material is a dielectric, then generally we do not want to change the first-surface (polarized) hemispherical reflectivity. The depolarized diffuse contribution to the hemispherical reflectivity, then, is the difference between the total reflectivity and the (small) first-surface contribution:

$$R_{diffuse}^{new}(0) = \text{DHR}(0) - R_{specular}^{integrated}(0) \tag{4.91}$$

Usually the first-surface specular reflectance is much less than the diffuse volumetric contribution for a dielectric. We rescale the two diffuse parameters equally to achieve the measured hemispherical reflectivity value:

$$rescale = \frac{R_{diffuse}^{new}(0)}{R_{diffuse}^{integrated}(0)} \tag{4.92}$$

Thus:

$$R_x^{new} = rescale \cdot R_x^{old} \tag{4.93}$$

$$R_{hov}^{new} = rescale \cdot R_{hov}^{old} \tag{4.94}$$

When the measured hemispherical reflectivity DHR(0) is less than the first-surface integrated value alone, we rescale both the specular and the diffuse equally to meet the new DHR(0). If the material is a clear dielectric (i.e., not a paint or pigmented plastic), then we should rescale both the specular and the diffuse equally to meet the new DHR(0).

If the material is a metal, then we rescale both the specular and diffuse BRDFs equally. The reasoning here is that fundamentally the Fresnel reflectivity drives the energy balance. There is no independent *volumetric* reflection process. So, if the total reflectivity at a different wavelength is higher, say, then both the specular and diffuse contributions must go up proportionally, since both are controlled at the microscopic level by Fresnel effects.

We rescale by the ratio of new to old total reflectivities as:

$$rescale = \frac{\text{DHR}(0)}{R_{specular}^{integrated} + R_{diffuse}^{integrated}} \tag{4.95}$$

We then apply this rescaling factor first to the bi_factor we interpolated:

$$\text{bi\_factor}_{new}(\theta_N) = \text{bi\_factor}_{old}(\theta_N) \cdot rescale \tag{4.96}$$

Equation 4.96 implies that every tabulated value for the bi_factor must be rescaled. Finally, we rescale the diffuse parameters.

Note that if we are working a problem at a single wavelength (e.g., active laser illumination) where we have measured data, then whatever method we use to rescale (and therefore renormalize the BRDF) should return exactly the measured parameters, unless the measured

DHR(0) is slightly different from the extracted BRDF parameters. This is usually the case. Since the independently measured DHR(0) is inherently more accurate, because it collects all the energy from the hemisphere (and for reasons of consistency), we always renormalize the BRDF model to the DHR(0).

In Figure 4.12 we render a typical BRDF and scattered intensity in 3-D as an iso-intensity surface. Notice the spread of the BRDF near the plane of the material. This is the usual turn-up of the BRDF at grazing angles where the denominator of the specular BRDF explodes because of the $\cos(\theta_r)$ term. On the right we rendered the intensity profile. The cosine in the numerator now cancels this turn-up, as expected.
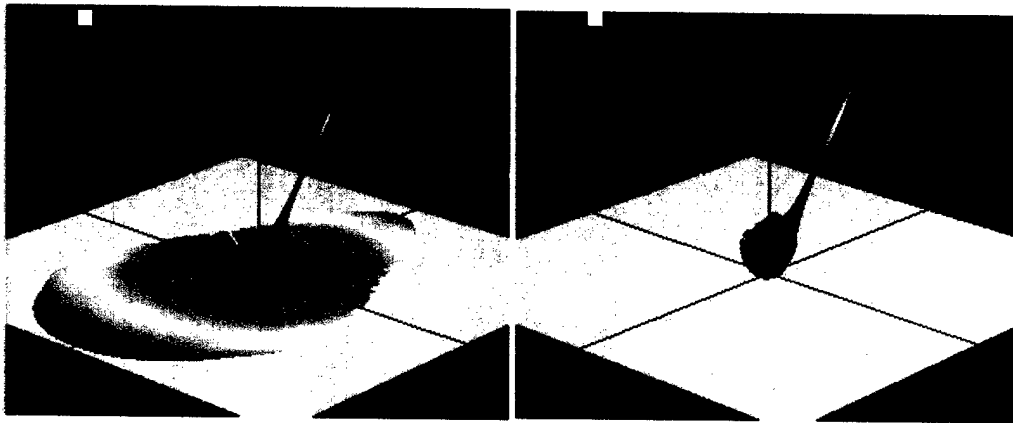
Figure 4.12. Rendered 3-D BRDF; the image on the left is the BRDF; the yellow box is the laser Source; the image on the right is the cross-section (intensity).

The lower intensity (blue) colors are due to the diffuse BRDF contribution. The yellow and red colors show the specular BRDF contribution. Figure 4.13 shows the same BRDF but with the functional values displayed on a hemispheric dome.
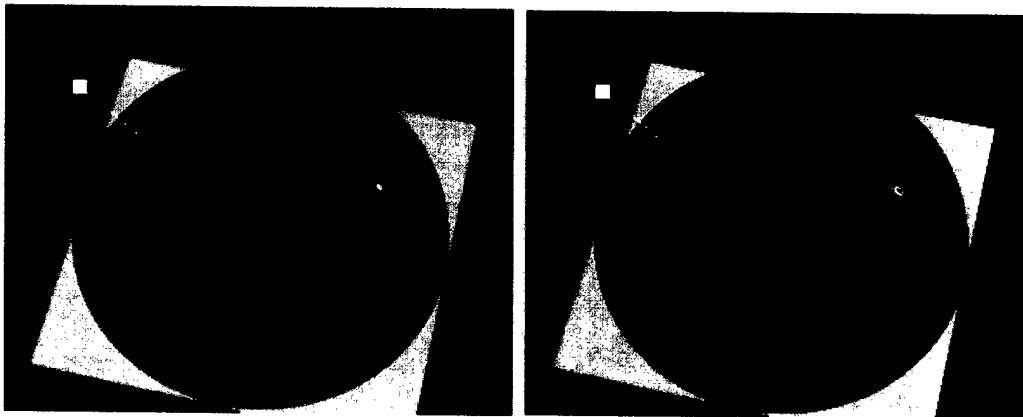
Figure 4.13. Same BRDF and Cross-sections Displayed on a Hemisphere

One of the interesting effects of scattering at grazing incidence is how the BRDF will become narrower along the cross-plane direction. The narrowing of the BRDF is caused by the rapid increase in the bi-section angle (θ_N) motion due to the optical gain $\delta\theta_N = \dfrac{\delta\theta_{LOS}}{2\cdot\cos(\beta)}$. Since β is near 90 degrees, the change in the bi-section angle is large for even small observer motions cross-plane. At the same time the BRDF is getting narrower, there is a cosine in the specular BRDF denominator that is also increasing its magnitude. The combined effect when integrating the total power is to keep it constant.
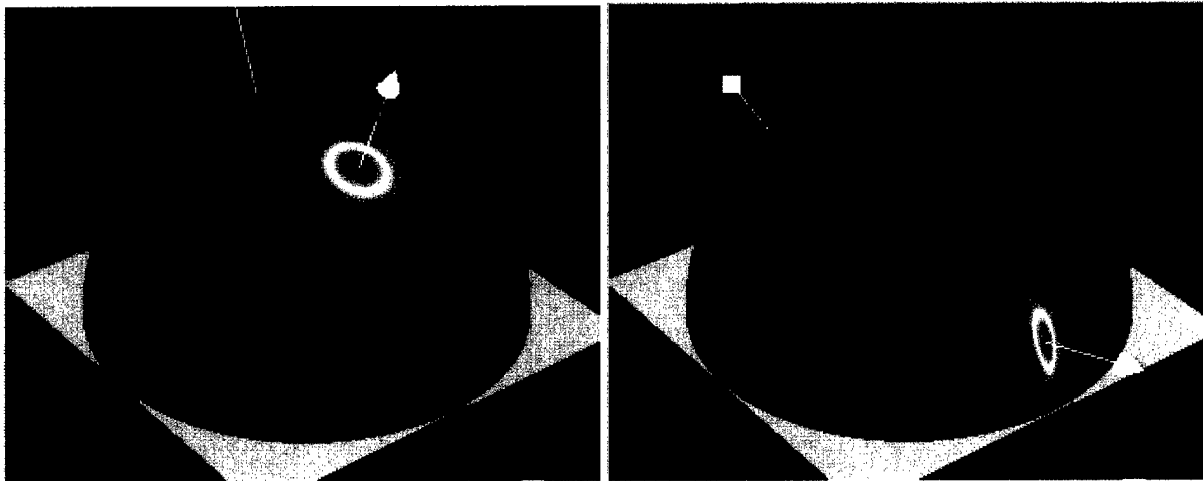
Figure 4.14. Typical BRDF at Near Normal Incidence on Left, and Grazing at Right

Figure 4.15 demonstrates the growth of the BRDF with increasing angle of incidence. In this case, the normalized Fresnel curves do not change very much, so the growth is caused primarily by the cosine in the denominator.
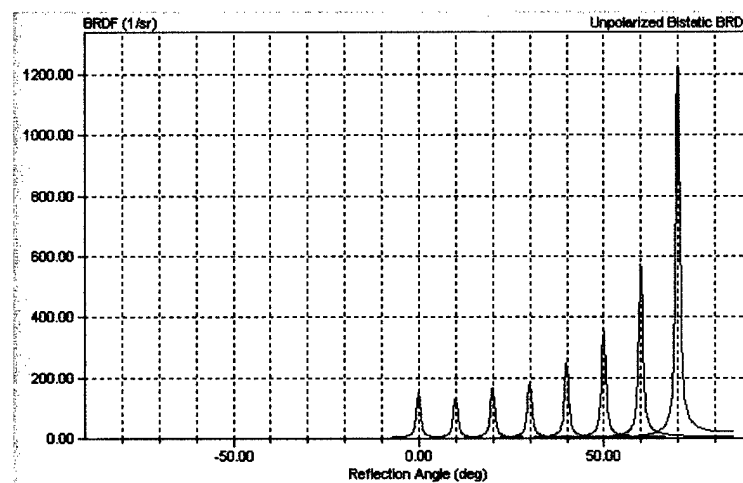
Figure 4.15. Growth of the BRDF Peak with Increasing Angle of Incidence

The usual objective of radiometric signature calculations is to predict the amount of power entering a receiver aperture, having scattered off a target body. The equations are rather simple, and they all stem from rearranging Equation 4.25. The equations that follow relate to small incremental surface areas that are flat and uniform. Sometimes when large areas are relatively flat, the smallness restriction can be lifted. We will also be interested in the definition and usage of the optical cross-section (defined differently from radar cross-section).

The lowest denominator in Equation 4.25 is defined as the incident irradiance. Consider a collimated beam with irradiance I_o (Watts/m^2) that is incident on a flat surface element δA (m) at an angle θ_i. The total power intercepted by the surface element is:

$$\delta P_i = I_o \cdot \delta A \cdot \cos(\theta_i) \qquad \text{(Watts)} \qquad (4.97)$$

The denominator can be better expressed by:

$$\delta P_i / \delta A = I_o \cdot \cos(\theta_i) \qquad \text{(Watts/m}^2) \qquad (4.98)$$

We now solve Equation 4.25 for the power collected by the receiving aperture δP_r:

$$\delta P_r = I_o \cdot \cos(\theta_i) \cdot \rho' \cdot \delta A \cdot \cos(\theta_r) \cdot \delta\Omega_{rec} \qquad \text{(Watts)} \qquad (4.99)$$

We recognize the rightmost three terms as an Etendue' - or the configuration factor:

$$\delta\xi = \delta A \cdot \cos(\theta_r) \cdot \delta\Omega_{rec} \qquad \text{(m}^2\text{-sr)} \qquad (4.100)$$

where the receiver aperture solid angle is:

$$\delta\Omega_{rec} = \frac{\delta A_{aperture} \cdot \cos(\theta_{LOS})}{Z^2} \qquad \text{(sr)} \qquad (4.101)$$

and $\delta A_{aperture}$ is the aperture collecting area, θ_{LOS} is the angle between the aperture pointing line of sight (LOS) and the line to the target (θ_{LOS} is always assumed zero), and Z is the range to the target. We can also introduce the concept of radiance, $J = I_o \cdot \cos(\theta_i) \cdot \rho'$ (Watts/m^2-sr) and express the received power very compactly as:

$$\delta P_r = J \cdot \delta\xi \qquad \text{(Watts)} \qquad (4.102)$$

Since the BRDF is directionally dependent, we sometimes refer to J as the directional radiance. Equations 4.99 or 4.102 are the fundamental radiometric relations we need to make radiometric predictions. Because targets are so complex, we must code these equations and use ray-tracing techniques to sample the target shapes, and must do the target geometry incrementally.

We need to define another important radiometric quantity called the optical cross section χ, as the ratio:

$$\chi = \frac{\delta P_{rec}}{I_o \cdot \delta\Omega_{rec}} = \rho' \left(\theta_i, \theta_r, \phi_r\right) \cdot \delta A \cdot \cos\left(\theta_i\right) \cdot \cos\left(\theta_r\right) \quad \text{(m}^2\text{/sr)} \qquad (4.103)$$

where we have rearranged Equation 4.99 to separate the target-specific attributes. The cross section is strictly a function of the target shape, its material properties, the illumination geometry, and the viewing geometry, and nothing else. When the irradiance is not uniform over the entire target, then the measured or apparent cross-section will become a function of the irradiance profile. In this case we would need to include the unit-irradiance beam profile into the calculation of the incremental cross-section contributions.

If we know the cross-section, we can always invert Equation 4.103 to predict the power at the receiver:

$$\delta P_{rec} = \chi \cdot I_o \cdot \delta\Omega_{rec} \quad \text{(Watts)} \qquad (4.104)$$

where I_o now represents the peak on-axis irradiance. The scattered intensity at the receiver aperture (without atmospheric losses) is:

$$I_{rec} = \frac{\delta P_{rec}}{\delta A_{rec}} = \chi \cdot \frac{I_o}{Z^2} = \rho' \left(\theta_i, \theta_r, \phi_r\right) \cdot \delta A \cdot \cos\left(\theta_i\right) \cdot \cos\left(\theta_r\right) \cdot \frac{I_o}{Z^2} \quad \text{(w/m}^2\text{)} \qquad (4.105)$$

4.4 Cross Section Radiometrics

For complex target scattering geometries, in practice we can calculate the total cross-section by integrating (summing) over many small differential surface area elements:

$$\chi_{total} = \frac{\sum \delta P_r}{I_o \cdot \Omega_{rec}} = \sum \rho' \cdot \delta A \cdot \cos(\theta_i) \cdot \cos(\theta_r) \qquad \text{(m}^2\text{/sr)} \qquad (4.106)$$

or, put into integral form:

$$\chi_{total} = \int \rho' \cdot \cos(\theta_i) \cdot \cos(\theta_r) \cdot dA \qquad \text{(m}^2\text{/sr)} \qquad (4.107)$$

Now, going back to Equation 4.105, the intensity incident at an observer is:

$$I = \tau_{atm} \cdot \chi_{total}(i, j) \cdot \frac{I_o}{Z^2} = \tau_{atm} \cdot \rho' \cdot \delta A \cdot \cos\left(\theta_i\right) \cdot \cos\left(\theta_r\right) \cdot \frac{I_o}{Z^2} \qquad \text{(w/m}^2\text{)}$$

$$(4.108)$$

where we now have included the possibility of some atmospheric transmission loss. Given the irradiance (assume for now its uniform) I_o along some incident direction, the observer is located a distance z along some other direction, and we know the total target cross section along the respective observer direction (by indexing into the cross section sphere), we can use Equation 4.108 to determine the intensity at that observer. We will cover atmospheric attenuation as a function of path direction in a later section. How we extend Equation 4.108 to dynamic motions

we will also cover later. The key algorithm we need now is how to efficiently calculate the total target cross section $\chi(i,j)$ for every "pixel" on the cross section sphere.

There are several ways to approach estimating the total scattered cross sections on the cross section sphere. At each pixel on the cross section sphere we need to know the total (integrated) cross section scattered from the *entire* visible surface area over the target. It would be nice if we could simply integrate over the target surface as a function of incident beam direction and pixel viewing direction. In general, since target shapes are so complex this realistically is just not possible. However, we can break the target down into its primitives and try to integrate on that level. In fact we will loop over each primitive in the model, evaluate the scattered cross sections from each one and accumulate with all others on the sphere. In addition we will separate diffuse from specular reflections to minimize computations by better matching the angular resolutions with the angular detail in the BRDFs.

Perhaps the most obvious approach would be to use our ray-tracing mechanics to first do the obstruction testing, then reflect the ray and find where on the cross section sphere it intersects. We expect that if the small incremental area occupied by the collector point were flat that a reflected spot ought to be painted around the specular ray direction on the cross section sphere. By meshing the target into a large number of small facets (i.e., collector points), we can, in the limit of differential surface areas, approach the true distribution of scattered cross sections for curved surfaces, and do flat surfaces with minimal effort. The obvious problem with this comes from having to mesh so many small facets on curved surfaces, especially if the target BRDF is at all shiny. The effect of not meshing finely enough is the appearance on discrete spots instead of smooth continuous distributions of light on the cross section sphere (and in the scenario scene as well). This is yet another offensive manifestation of aliasing. So, we do not want to simply shoot and bounce rays onto the cross section sphere, unless the primitive surface is a plane.

A better approach to evaluating the cross sections for a curved surface is to first realize that for singly (e.g., cylinders) or doubly (e.g., sphere) curved shiny surfaces, glints move smoothly as the geometry changes. In fact, for the analytic primitives used in the CAD models, we can integrate the specular scatter as a initialization step. For unobstructed primitives, we can quickly and accurately estimate the cross section as a function of geometry. When parts of a surface are obstructed by other primitives, we can reduce the scatter in proportion to the number of collector points obstructed. Therefore, by doing a pre-calculation we can avoid severe aliasing artifacts caused by doing a finite meshing of the target surface area.

4.5   Plate Cross Section

The plate, unlike all the other primitive shapes, has no curvature. A glint only occurs when the bi-section vector $\hat{H}$ (which bi-sects the laser source direction and an observer direction) is aligned with the singular surface normal. If the surface is very shiny, the peak cross-section is driven by the peak of the BRDF and is found along the ideal specular direction. If it is moderately shiny, then the peak can occur at angles slightly lower than the specular direction. In any event we will need to paint the reflected spot on the cross section sphere out to some angle where the BRDF has dropped sufficiently.

In general, the cross section for a flat surface of area δA is:

$$\chi = \rho'_{specular} \ (\theta_N, \beta) \cdot \delta A \cdot \cos (\theta_i) \cdot \cos (\theta_r) \qquad (\text{m}^2/\text{sr}) \qquad (4.109)$$

where the bisecting vector $\hat{H}$ makes an angle θ_N with the surface normal, and the angle between either the incident for reflected LOS and the bisecting vector is β. Recall from the discussion of BRDFs that the specular functional form for the MB BRDF is:

$$\rho'(\theta_N, \beta) = \frac{R_{un}(\beta)}{R(0)} \cdot \frac{bi\_factor(\theta_N)}{\cos(\theta_i) \cdot \cos(\theta_r)} \cdot S_o(\theta_N, \beta) \qquad (\text{sr}^{-1}) \quad (4.109)$$

hence we can combine Equations 4.108 and 4.109 such that:

$$\chi = \frac{R_{un}(\beta)}{R(0)} \cdot bi\_factor(\theta_N) \cdot S_o(\theta_N, \beta) \cdot \delta A \qquad (\text{m}^2/\text{sr}) \qquad (4.110)$$

The shadow/masking term S_o is usually near 1 for most materials, or for small θ_N. The leading normalized Fresnel term is also near 1, except at the larger grazing angles. The plate specular cross section then becomes:

$$\chi = F(\beta) \cdot bi\_factor(\theta_N) \cdot \delta A \qquad (\text{m}^2/\text{sr}) \qquad (4.111)$$

When the BRDF is Lambertian (i.e., $\rho' = \dfrac{R_{diffuse}}{\pi}$), then the diffuse cross section for a plate is:

$$\chi_{plate} = \left[\frac{R_{diffuse}}{\pi} \cdot \delta A \cdot \cos(\theta_i) \right] \cos(\theta_r) \qquad (\text{m}^2/\text{sr}) \qquad (4.112)$$

where $R_{diffuse}$ is the diffuse fraction of the hemispherical reflectance. The term in brackets is a function of the illumination incident direction. The diffuse cross section falls off as a cosine of the observing angle. So, for every incremental area along the length of the cylinder, we get a scattered cross section that is a function of the observing angle, which in turn is a function of the local surface normal.

4.6 Cylinder Cross Section

The cylinder has a long linear glint only when the bi-section vector is at right angles to the cylinder symmetry or rotational axis. Consider three points, the laser source position, the missile target position, and an observer position, all coincident with some plane. Let $\hat{H}$ be a bisecting vector between the LOS to the laser from the target, and the LOS to the observer from the target. In principle, the cylinder could be oriented in any manner relative to this plane. A convenient geometry for the cylinder is displayed in Figure 4.16. The z-axis is vertical in the figure, and is coincident with the bisecting vector $\hat{H}$, when the light source and observer are perpendicular to the rotational axis of the cylinder. In general, however, the $\hat{H}$ bisecting vector is not coincident

with the cylinder z-axis, but forms some angle γ to it, as shown in Figure 4.17. The $\hat{H}$ bisecting vector is always in the y-z plane. The x-axis is to the right, and the y-axis is into the page.
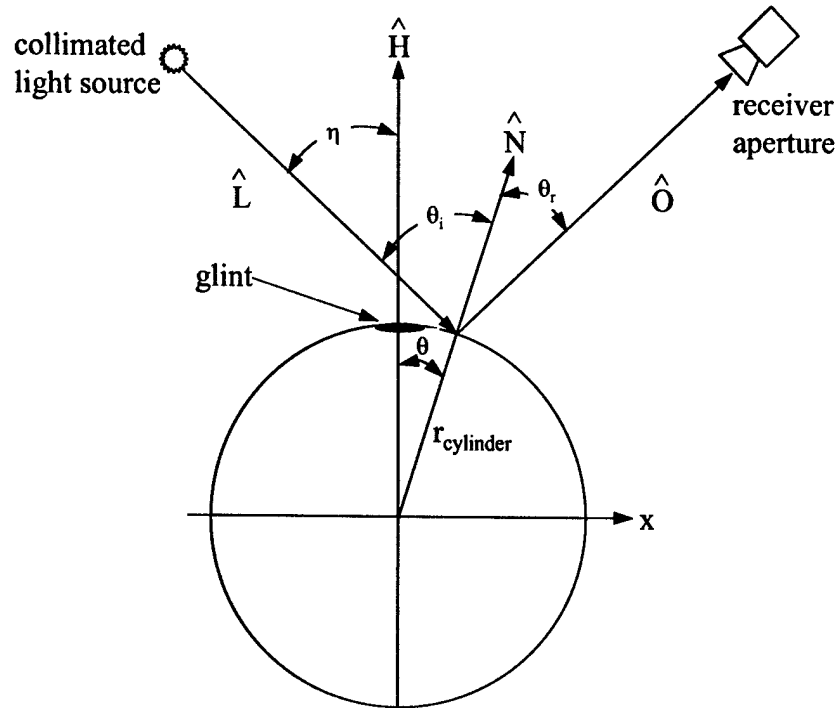
Figure 4.16. Geometry Used for Cylinder Analysis, Looking Along the Rotation Axis

Assume the cylinder length is L, along the y-axis. The glint, by symmetry, is at the top of the cylinder where θ is zero. The integration for the total cross section is done in polar coordinates, with θ measured down from the z-axis. Recall that both the light source and observer are at large distances from the cylinder such that the ray directions are essentially the same when we integrate along the length of the cylinder.

$$\chi_{cylinder}(\eta,\gamma) = \int_{-L/2}^{+L/2} dl \cdot \int_{-\pi/2+\eta}^{\pi/2-\eta} \rho' \cdot \cos(\theta_i) \cdot \cos(\theta_r) \cdot r d\theta \quad \text{(m}^2\text{/sr)} \quad (4.113)$$

where r is the radius of the cylinder, and η is the angle the projected incident vector makes with the z-axis in the x-z plane. As the included angle between the source and observer (i.e., η/2) increases from zero, the observer sees less of the illuminated cylinder body, the laser illuminates further around the side, and thus the limits of the integral should reflect this, and the cross section for grazing incidence should therefore be less than for mono-static illumination.

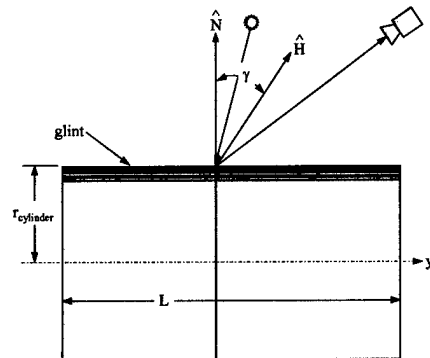The surface normal vector for a cylinder as a function of polar angle θ is:

$$\hat{N} = \sin(\theta) \cdot \hat{i} + \cos(\theta) \cdot \hat{k} \quad (4.114)$$

where $(\hat{i}, \hat{j}, \hat{k})$ are three orthogonal unit vectors along the respective coordinate axes. The bisection vector is constant and points along:

$$\hat{H} = \sin(\gamma) \cdot \hat{j} + \cos(\gamma) \cdot \hat{k} \qquad (4.115)$$

As γ increases from zero, the observer sees less and less of the glint at the top of the cylinder.

Figure 4.17. Cylinder Geometry as Seen From the Side
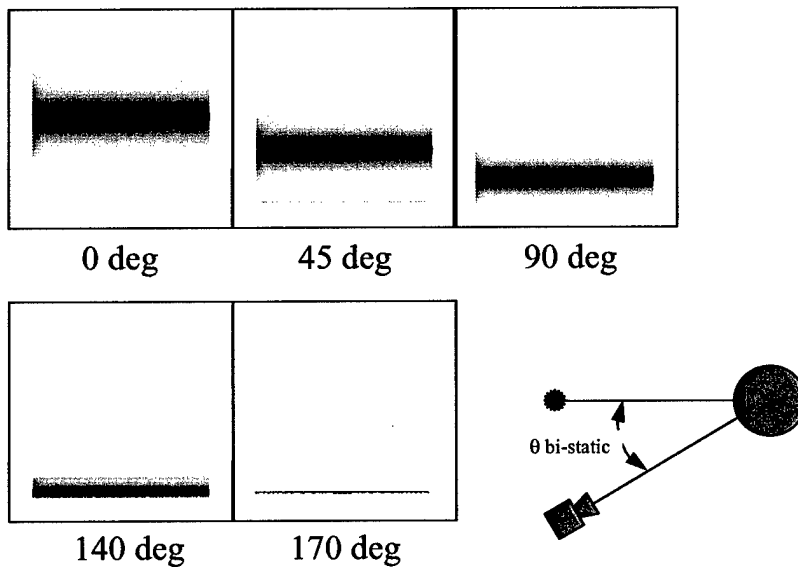
Figure 4.18. Rendered Views of a Cylinder as a function of bi-static angle.

The cylinder is rendered from the side, while the graphic on the right defines the bi-static angle θ noted under each rendered view. The viewer is located in the horizontal plane while the source is moved downward.

Figure 4.18 shows rendered side-views of a cylinder where the bistatic angle is increased from zero to 170 degrees. From 0 degrees to 90 degrees the observer can see almost all of the energy

reflected (i.e., the entire glint). From 140 to 180 degrees the observer really can not see all of the scatter because the cylinder body self-obstructs. In our development, we will essentially ignore this self obstruction effect. The error is on the conservative side because we will assign a cross section higher than it really should be.

The angle θ_N (micro-facet tilt angle) used as one argument for the BRDFs is given by:

$$\theta_N = \cos^{-1}[\hat{H} \circ \hat{N}] = \cos^{-1}[\cos(\theta) \cdot \cos(\gamma)] \tag{4.116}$$

When $\gamma = 0$ (glint condition), then $\theta_N = \theta$ which is the variable of integration. We are interested in the integrated cross section due to the specular reflection alone. If we assume the material is reasonable shiny, then we can extend the integration limits to a full half circle:

$$\chi_{cylinder} = L \cdot \int_{-\pi/2}^{\pi/2} \rho'(\theta_N, \beta) \cdot \cos(\theta_i) \cdot \cos(\theta_r) \cdot r d\theta \qquad \text{(m}^2\text{/sr)} \tag{4.117}$$

If not all of the cylinder length is illuminated (i.e., partially obscured by other target components), then the length L can be reduced appropriately. The ray tracing will determine what parts of each primitive will be obstructed.

As a pre-calculation step we will remove the Fresnel term from the integral because its nearly 1 and because its constant (i.e., β is constant) over the integration. The cosine terms cancel and we are left with:

$$\chi_{cylinder} = F(\beta) \sum_i \delta L \cdot r \int_{-\pi/2}^{\pi/2} bi\_factor \left[\cos^{-1}(\cos\theta \cdot \cos\gamma) \right] \cdot d\theta \qquad \text{(m}^2\text{/sr)} \tag{4.118}$$

where $F(\beta)$ is the normalized Fresnel term, and we have explicitly shown the summation over those incremental lengths of the cylinder that are illuminated. Given one of the material BRDFs, we can pre-integrate the bi_factor as a function of the angle γ and store in an array:

$$G_{cylinder}(\gamma) = 2 \int_0^{\pi/2} bi\_factor \left[\cos^{-1}(\cos\theta \cdot \cos\gamma) \right] \cdot d\theta \qquad \text{(m}^2\text{/sr)} \tag{4.119}$$

While evaluating the scattered cross sections, we will find the angle γ and interpolate this function, and finally include the remaining terms:

$$\chi_{cylinder} = F(\beta) \sum_i \delta L \cdot r \cdot G_{cylinder}(\gamma) \qquad \text{(m}^2\text{/sr)} \tag{4.120}$$

Notice that in Equation 4.119 that the larger the angle γ, the less the bi_factor that gets integrated, and hence the smaller the $G(\gamma)$ factors. This is a simple manifestation of being out of the glint geometry. We therefore only evaluate the $G(\gamma)$ factors for γ angles from 0 to where the specular

BRDF drops to some low value. We therefore need to evaluate each specular BRDF to determine the width of its specular lobe.

When the BRDF is only moderately shiny (say 20 degrees wide or more), the limits of the integration in Equation 4.119 become more important because the glint at grazing angles reduces. However, to simplify the calculations, and keep things conservative, we do not reduce the limits of integration. We use the maximum $G(\gamma)$ factors one would see for mono-static illumination.

Notice that Equation 4.120 is valid for various radii and length cylinders. In fact, we can cache multiple cylinders at once if they have the same BRDF and are parallel (but not necessarily co-axial). As we paint the cross section sphere, we move from pixel to pixel, re-evaluating the angles γ and β.

The cross section of a diffuse cylindrical Lambertian scatter is completely analytic. But, the equations are not so simple when the diffuse scatter BRDF is not entirely Lambertian. A completely numerical approach is again useful here. While surface shape is important to the total scattered diffuse energy, local curvature effects are not as important as for shiny surfaces. Diffuse scatter is not very directional. Now we can easily approximate curved surfaces, like a cylinder, by faceted surfaces. This is especially important for doubly curved surfaces like a sphere or ellipsoid where the equations are definitely not analytic.

We use the local collector points to define the faceted areas. Each column of collector points defines a long facet plane whose area is:

$$\delta A = L \cdot r \cdot \Delta \theta \qquad \text{(m}^2\text{)} \qquad (4.121)$$

where $\Delta \theta$ is the angular separation of the collector points.

We do separate bookkeeping on separate cross section spheres, for the specular and diffuse components. The diffuse cross sections are calculated in two steps. The first step is to use Equation 4.121 to accumulate a single cross section value along the direction of the local surface normal, at a pixel on a low resolution cross section sphere. This is done for all small areas on the entire target surface. The second step then is to take each accumulated pixel value on the sphere and distribute it over a hemisphere with the appropriate $\cos(\theta_r)$ weighting.

4.7 Cone Cross Section

The cone has a wedge-shaped glint only when the bisection vector is at right angles to the edge. Again, consider the same bisecting vector $\hat{H}$. In principle the cone could be oriented in any manner relative to the plane containing the source and observer LOS vectors. A convenient geometry for the cone is displayed in Figure 4.19. The z-axis is along the cone rotation axis. If the cone apex makes an angle δ, then the surface normal in the x-z plane also make this angle with the x-axis. Assume the cone has a height H_C, along z. the cone base radius is R_B. The cone apex angle therefore is $\tan(\delta) = R_B / H_C$. The edge length is $L = H_C / \cos(\delta)$.

The bisecting vector $\hat{H}$ is always in the y-z plane. In general, the $\hat{H}$ bisecting vector forms some angle γ to the surface normal, that can vary from 90-δ to 90+δ. However, beyond 90 degrees to the left none of the cone can be seen. We will use the polar angle θ for integrations around the cone.
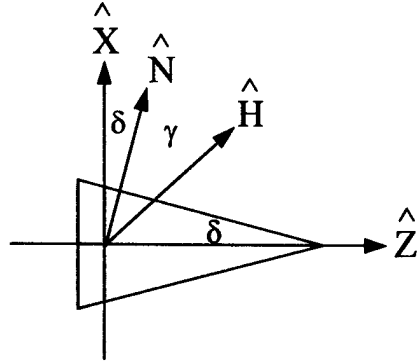
Figure 4.19. Side View of the Cone Geometry Showing an Off-Glint Bisecting Condition

The surface normal as a function of the polar angle θ (measure down from the x-z plane) is:

$$\hat{N} = \cos(\delta) \cdot \cos(\theta) \cdot \hat{i} + \cos(\delta) \cdot \sin(\theta) \cdot \hat{j} + \sin(\delta) \cdot \hat{k} \qquad (4.122)$$

The bi-section vector is:

$$\hat{H} = \cos(\delta + \gamma) \cdot \hat{i} + \sin(\delta + \gamma) \cdot \hat{k} \qquad (4.123)$$

with no component in the y-direction by definition. When γ is zero then the glint condition is met. The angle θ_N is given by the dot product:

$$\theta_N = \cos^{-1}[\hat{H} \circ \hat{N}] = \cos^{-1}[\cos(\theta) \cdot \cos(\delta) \cdot \cos(\delta + \gamma) + \sin(\delta) \cdot \sin(\delta + \gamma)] \qquad (4.124)$$

The cross-section integration for a specular BRDF is:

$$\chi_{cone}(\gamma) = \int_0^{H_c} \frac{h \cdot \tan(\delta)}{\cos(\delta)} dh \cdot \int_{\theta_{min}}^{\theta_{max}} \rho'(\theta_N, \beta) \cdot \cos(\theta_i) \cdot \cos(\theta_r) \cdot d\theta \quad (\text{m}^2/\text{sr}) \qquad (4.125)$$

where the lower and upper integration limits on θ depend on the angles the incident light and viewer LOSs make with the z-axis. In Figure 4.20 the cone is rendered as a function of bisecting angle.

0 deg **45 deg** **90 deg**

140 deg **170 deg**
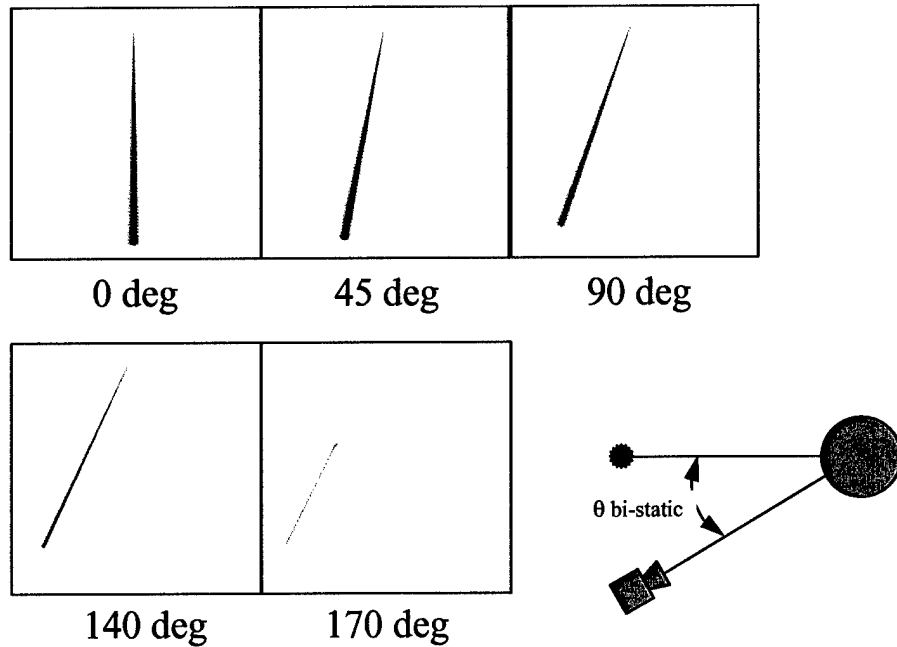
θ bi-static

Figure 4.20. Rendered Views of the Cone as a function of Bistatic Angle.

The cone is rendered from the side, while the graphic on the right defines the bi-static angle θ noted under each rendered view. The viewer and laser are located such that the bisecting vector H is always aligned with the local surface normal. The cone tends to rotate in each view.

The glint takes on a slanted angle because the sides of the cone are at an angle (cone apex is up).

$$G_{cone}(\gamma,\delta) = 2\int_{0}^{\pi/2} bi\_factor\left[\cos^{-1}\left(\cos\theta \cdot \cos\delta \cdot \cos(\delta+\gamma) + \sin\delta \cdot \sin(\delta+\gamma)\right)\right] \cdot d\theta \quad \text{(m}^2\text{/sr)} \quad (4.126)$$

While evaluating the scattered cross sections, we will find the angle γ and interpolate this function, and finally include the remaining terms:

$$\chi_{cone} = F(\beta)\sum_{i} \frac{(h_2^2 - h_1^2) \cdot \tan\delta}{2\cos\delta} \cdot G_{cone}(\gamma,\delta) \qquad \text{(m}^2\text{/sr)} \qquad (4.127)$$

where $h_2-h_1=\Delta h$, the vertical separation between collector points (along the z-axis), and h_x is the distance the row of collector points is from the vertex point.

The diffuse scatter component is treated in the same manner as for a cylinder.

4.8 Beam Coupling

In the cross section equations above, we assumed the illumination was constant for every differential surface area. Recall that we actually meshed each primitive surface with collector points that not only define the ray trace aimpoints for obstruction testing, but each collector point

also serves to represent a small patch of surface area. When we do the obstruction ray tracing, we first determine if the laser beam has a LOS access to each collector point patch. If it does, then we will evaluate the diffuse scatter (regardless of any further out-going obstructions) into the hemi-space above the local surface patch. If the specular reflected direction is unobstructed, then we mark that area element as contributing to specular reflection calculations. When we do the radiometrics, we in effect add each of these unobstructed area elements together for groups of collector points whose surface normal are aligned parallel (e.g., along the length of a cylinder).

If the user selects a flood-load beam profile, then each collector point area element is equally weighted with a unity coefficient. However, if the user selects a Gaussian beam profile, then during the ray-tracing step, we mark each unobstructed collector point with a relative beam intensity coefficient. Thus surface areas near the peak of the beam are weighted more heavily than those further away. This not only couples the beam profile into the cross section distributions, but it also emphasizes those parts of the target that are illuminated. Thus for example, if a small beam is put on the main cylinder body of a missile, then the scattered cross sections will form a ring around the target. But, if the beam is put on a tail fin, then the scattered cross section distribution will be a spot.

4.9 Painting Cross Sections Onto the Cross Section Sphere

We have discussed how the target model is represented geometrically by primitive shapes, and how to trace rays to small surface patches (collector points) to resolve obstructions. We then discussed the basic scattering radiometrics (i.e., the BRDF) for these small surface areas, and how to do local integrations of the scatter to arrive at the effective radiometric cross section for different surface shapes. Our next objective is to build a complete cross section sphere for each discrete time step of the simulation, as the target moves along its trajectory path.

A target model usually contains a number of primitive shapes (a missile would at least be a body cylinder and a cone at its nose), each of which contributes, in turn, to the net cross section. Therefore we will loop over all the primitives in the model and evaluate the contributions of each primitive in turn. How the primitives contribute depends of course on the respective primitive shape and curvature according to the equations we developed above, and its orientation. Now, for each primitive we separate the scatter into two components, one specular and the other diffuse.

The diffuse scatter first goes to a single pixel on a low resolution cross section sphere along the direction of the local surface normal. This is done for each group of collector points on a primitive surface where the local surface normal points along the same direction. Once all diffuse contributions are accumulated along all normal vector directions, each non-zero pixel is then distributed over a second hemisphere with the same (low) resolution with the cosine fall-off. This is done at lower resolution because diffuse scatter is second order in importance, and because the distribution varies so slowly. We do not consider target model self obstructions to the outward diffuse scatter because they are so much lower in intensity, and because it would be prohibitive to check all the possible scatter directions into a hemisphere (with ray tracing) for every small surface patch area. This again errs on the side of conservatism because we are potentially allowing scatter to exist where it might actually get obstructed.

The specular scatter nominally goes along the reflection ray direction, the directionality depending on the BRDF shape. If the material was perfectly specular we could just follow the specular reflected rays and add contributions to individual cross section sphere pixels. However, few materials really have a BRDF distribution that specular. Instead we will need to do more work to spread the scattered cross sections to neighboring pixels according to the equations we developed above. We do know that the peak scattered cross sections probably need to be assigned to the specular directions. Lesser cross section values need to be assigned to adjacent pixels, down to a point where there are negligible values.

The simplest approach to *painting* the scattered cross section values onto the cross section sphere would be to loop over *every* pixel on the cross section sphere, evaluate the scattered contribution toward that pixel from *every* small area of every primitive. While this approach would be simple and thorough, the computational cost would be prohibitive. The cross section sphere contains close the two hundred thousand pixels, and most will capture little or no energy! A more efficient approach is to follow the path of significant scattered intensities and never visit pixels far from any specular energy.

The primitive shapes can be classified into three types, flat surfaces, singly curved surfaces (cylinders, cones), and doubly curved surfaces (spheres, ellipsoids). For flat surfaces we follow the specular reflected ray to where it intercepts the cross section sphere. The peak scatter will go here. We then visit all the surrounding pixels, out to some radius, and evaluate the scatter cross section until the contributions become insignificant. A spot will result.

When the primitive shape is singly curved, we need to again follow the reflected ray directions, only in a ring around the primitive. This ring covers a large number of pixels, every one of which would get a peak specular value. We then walk along the rows to either side of the peak pixel and reevaluate the scattered cross section with the off-specular geometry angles until the contributions are again insignificant.

We showed in two dimensions how specular light reflects from a cylindrical shape in Figures 1.3 and 1.4 in the first section. Again, the trick is to visit only those pixels that have significant potential for scattered light according to the BRDF. We also showed a simplified low resolution view of how the cross section sphere is organized. Imagine a ring of collector points around a cylinder (200 in fact). Half the points are obstructed on the far side of the surface and hence contribute nothing. If we follow a ray from the laser source to each visible point, reflect it, and follow the reflected ray to the sphere (now in direction cosine space), we would intercept one pixel on the cross section sphere. We get the same pixel for any surface normal parallel to the one in question. If we follow a ray for the next collector point in order around the cylinder, then we probably hit a new pixel on the cross section sphere, perhaps on a new row and column.
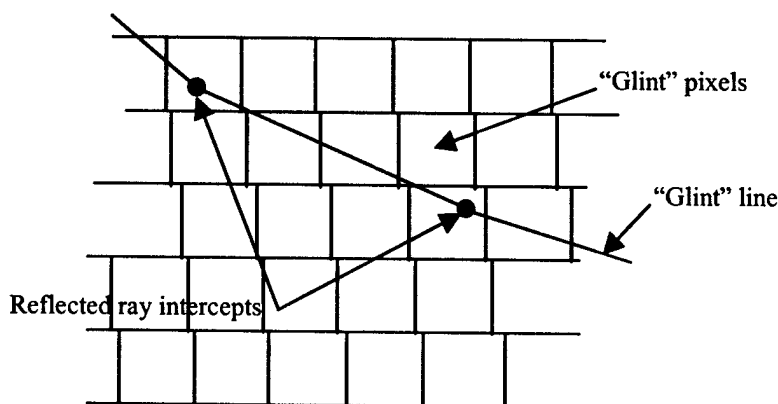
Figure 4.21. Small Section of the Cross Section Sphere Showing Glint Pixels

If we connect these two intercepts with a line and find all those pixels this line passes through, we can at least identify where the most intense reflected light should be evaluated. In Figure 4.21 two reflected ray intercepts are shown as green dots. The pixels pierced by the line connecting them are identified as "glint" pixels and colored green. We can follow this "glint line" all the way around the cross section sphere, and thus identify all pixels where we expect a maximum reflected cross section (or intensity later). While there is no clear cut column geometry on the cross section sphere, the rows do maintain a constant latitude all the way around the sphere. Most rows will have places where the glint line passes through it twice.
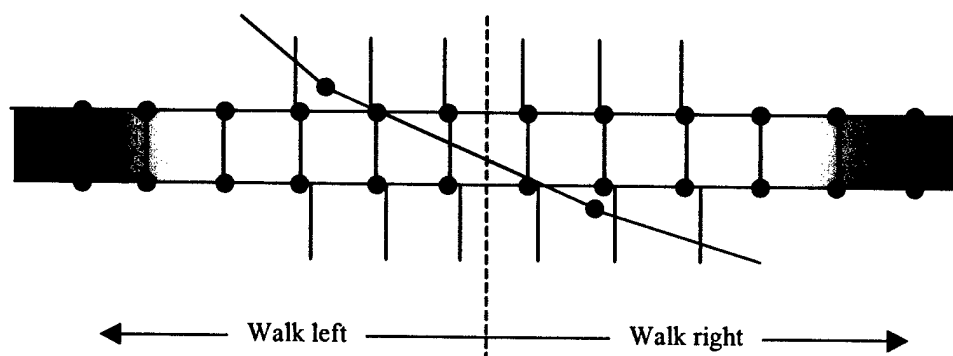
Figure 4.22. A Single Row of Pixels is Evaluated.

Consider the row of pixels shown in Figure 4.22. With the glint line defining the center and brightest part of the reflected intensity pattern, we walk the algorithm left and right from this point, painting the cross section values into the respective pixels. In fact we evaluate the reflected cross section at the four corners (red dots) of each pixel and pick the maximum value to represent the whole pixel area (solid angle). For each corner evaluated, we need to construct the bi-section vector, look-up the G(γ) factors, and determine the effective cross section.

The above discussion applies for a cylinder or a cone. For a plate the same basic concept holds but there is no glint line as such. A single intercept point is identified via the ray trace, and the algorithm visits enough rows, above and below the intercept point, to adequately capture the

reflected spot. Again, the algorithm walks left and right until the scatter values are low enough to stop traversing.

The shading algorithm for a cylinder, a cone, and a plate are therefore all slightly different.
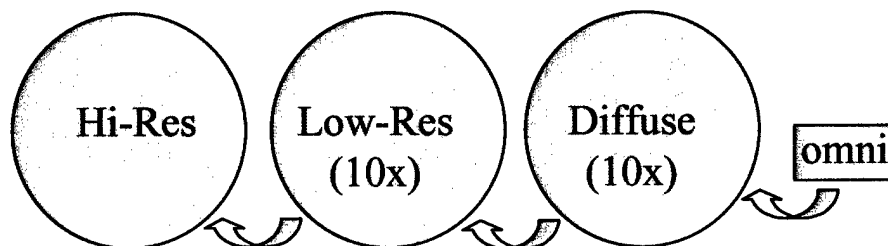
Figure 4.23. Final Step of Accumulating Cross Sections From Low to High Resolution

The final step, once each cross section sphere is complete at its respective resolution, is to accumulate the cross sections. Starting at the lowest resolution, we resample the pixels values of the lower resolution sphere to the next higher resolution sphere and accumulate cross sections. This is done up to the highest resolution, and final sphere, on the left in Figure 4.23. At this point the lower resolution spheres are discarded, and only the highest resolution sphere is kept for further processing. This final sphere represents the net scatter by all shape types, and all BRDFs both specular and diffuse, at one time step in the scenario.

4.10 Atmospheric Attenuation

As light propagates through the atmosphere it is both absorbed and scattered by molecules and large aerosols particles. The net effect is a loss of total power in a beam (or ray) with distance traveled. We need to model the attenuation affect of the atmosphere for an arbitrary ray path in the LRST code. This is necessary for the laser beam propagation from source to target, and from all reflected ray paths to either the ground or space.

As inputs to the LRST code we will need to provide tables of extinction coefficients as a function of altitude for different atmospheric conditions, by laser wavelength. The code will then need to calculate the point to point attenuation for light as rapidly and accurately as possible. Especially in this case, the accuracy of the calculation need only approximate the actual losses. Typical atmospheric transmission codes like Hi-Tran provide generic total path extinction values, based on a lot of *typical* (measured) atmospheric parameters. However, real atmospheric transmissions are highly dependent on absolute local world location, latitude, weather conditions, season, volcanic event history, time of day, propagation direction, and many other factors that can change on a daily basis, or even by the hour. Thus, in the end we can only model the atmospheric loss effect, to some approximate accuracy. The most conservative condition of course is to assume no atmospheric losses at all. This is a necessary option. We will provide a simple extinction model that estimates the point to point loss of power, either in the laser beam or in the viewing direction, based on a supplied table of extinction values.

For the ABL laser wavelengths, the atmospheric attenuation is not a major effect except for a few isolated ray directions like along long nearly horizontal paths to space, or in low altitude humid conditions where the extinction length is unusually short. While including atmospheric attenuation increases the fidelity of the simulation, it does not increase the conservatism because it only reduces the intensities. For instance, if the user were to select and model a low visibility (i.e., significant extinction) atmosphere model and conclude there were no hazards, when in fact the actual atmosphere turns out to be very clear, then the possibility for missing an actual hazard would exist.

For a uniform distribution of atmospheric scatter with plane parallel waves propagating through it, the intensity I (w/m^2) of a beam after traveling a distance L through the atmosphere is attenuated to:

$$I(L) = e^{-\sigma \cdot L} \cdot I(0) \qquad \text{W/m}^2 \qquad (4.128)$$

where σ (m^{-1}) is some effective extinction coefficient. The extinction coefficient σ depends on wavelength, and the molecular and aerosol densities in the atmosphere. We will not attempt to model the origins of the extinction coefficients, only the net attenuation effect contained in Equation 4.128.

An approximate model typically used for the atmosphere is to assume vertically stacked uniform slabs of air mass where the densities of various species of particles are constant within a single slab. This leads to the approximation that the extinction coefficient $\sigma(h)$ is a simple function of altitude and holds constant along short horizontal paths. For very long horizontal distances the earth's curvature begins to deviate the altitude of the ray path and the constant density approximation breaks down accordingly. Figure 4.24 displays the simple short path atmosphere model geometry.
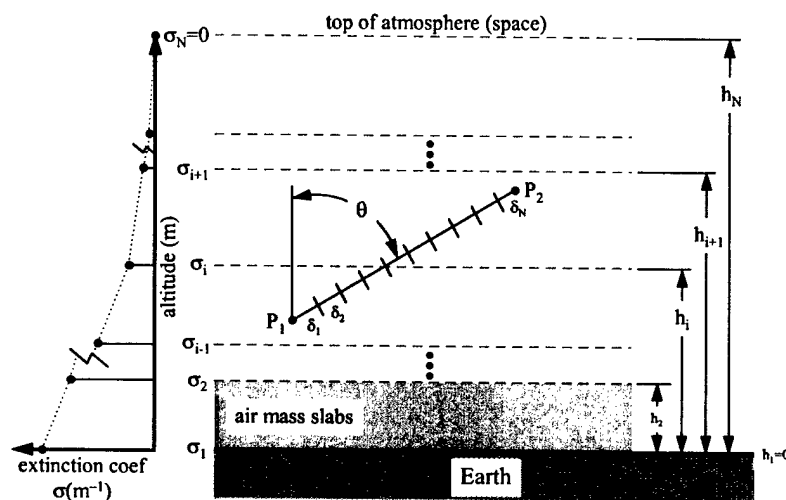
Figure 4.24. Uniform Stacked Flat Earth Atmosphere Model

At each altitude an atmospheric extinction coefficient is associated with it in table form $\{h_i, \sigma_i\}$. At the last altitude h_N, the extinction coefficient, by definition, is zero (i.e., at space). At ground level (i.e., h=0) a coefficient must be provided (otherwise the coefficient at the lowest altitude must be assigned to ground level). As one moves upward in altitude between enumerated levels, the extinction coefficient does not change stepwise, but rather it changes linearly between the two values specified at the bounding altitudes:

$$\sigma(h) = \frac{\sigma_{i+1} - \sigma_i}{h_{i+1} - h_i} \cdot (h - h_i) + \sigma_i \qquad \text{m}^{-1} \qquad (4.129)$$

where h is an altitude such that $h_i \leq h \leq h_{i+1}$. Obviously one would want to sample the table such that the extinction coefficients change reasonably smoothly. The higher the altitude, typically one can take longer steps because the atmosphere is less dense higher up.

In Equation 4.128 we assumed the atmosphere was isotropic. In general, even within a single slab the extinction will vary. Next we assume the atmosphere is isotropic along horizontal paths, but changes linearly in altitude. If one accumulates the attenuation along a skew path through the atmosphere going from one altitude to another, the result is an integration of the extinctions along that path. For example, along the path from point P_1 to P_2 (in Figure 4.24):

$$I(L) = e^{-\left(\sum \sigma_1 \cdot \delta_1 + \sigma_2 \cdot \delta_2 + \dots + \sigma_n \cdot \delta_n\right)} \cdot I(0) = e^{-\alpha} \cdot I(0) \quad \text{W/m}^2 \qquad (4.130)$$

where the distance increments δ_i are assumed small compared to the changes in the extinction coefficients. In the limit of infinitesimal step sizes:

$$\alpha = \int_{P_1}^{P_2} \sigma(x) \cdot dx = \left| \int_{\infty}^{P_1} \sigma(x) \cdot dx - \int_{\infty}^{P_2} \sigma(x) \cdot dx \right| \qquad (4.131)$$

notice the desired integrations are from space down to a point, and the absolute value is used because only the difference in optical density is significant. Since we have linearized the extinction coefficients in Equation 4.129 we can do the integrals easily. The integral from infinity to the top of the atmosphere (i.e., the last altitude where $\sigma_N = 0$) is zero. The integral from the top of the atmosphere to either point P is a series summation of integrals because the extinction function is only piecewise continuous (i.e., a sampled table of values). So we first integrate over each full discrete interval, between h_N to h_m, all the way up to the last interval that contains the point P, and then do the partial interval:

$$\alpha_{P_2} = \int_{\infty}^{P_2} \sigma(x) \cdot dx = \sum_{j=N}^{i+2} \int_{h_j}^{h_{j-1}} \sigma(x) \cdot dx + \int_{h_{i+1}}^{h} \sigma(x) \cdot dx \qquad (4.132)$$

where there are say, m full discrete intervals from the top of the atmosphere to the last partial interval, and h is the final altitude above mean sea level where $h_i < h < h_{i+1}$. To reiterate, the

integration is done from space downward, but the indexing starts at ground level. The differential path length $dx = \sec(\theta) \cdot dh$ is increased by $\sec(\theta)$ for slant directions.

The integral across any single discrete interval from altitude h_{j+1} to h_j is:

$$\alpha_j = \sec(\theta) \int_{h_{j+1}}^{h_j} \sigma(h) \cdot dh = \sec(\theta) \int_{h_{j+1}}^{h_j} \left[\frac{\sigma_j - \sigma_{j+1}}{h_j - h_{j+1}} \cdot (h_{j+1} - h) + \sigma_{j+1} \right] \cdot dh = \sec(\theta) \frac{(\sigma_{j+1} + \sigma_j) \cdot (h_{j+1} - h_j)}{2}$$

(4.133)

where the optical density α_j for each interval is positive. Note the linearization of the extinction profile.

The partial integral across any interval from altitude h_{j+1} to h is:

$$\Delta\alpha_i = \sec(\theta) \int_{h_{i+1}}^{h} \left[\frac{\sigma_i - \sigma_{i+1}}{h_i - h_{i+1}} \cdot (h_{i+1} - h) + \sigma_{i+1} \right] \cdot dh = \sec(\theta) \left[\frac{\sigma_i - \sigma_{i+1}}{2 \cdot (h_{i+1} - h_i)} \cdot (h_{i+1} - h)^2 + \sigma_{i+1} \cdot (h_{i+1} - h) \right] ($$

4.134)

If we have N altitude samples $h = \{h_1 = 0, h_2, h_3, ..., h_N\}$ then there are N-1 full nonzero intervals, each with a corresponding optical density value α_j given by Equation 4.133. The value of the highest interval (i.e., from h_N to ∞) is of course zero. The first interval (or air mass slab 1) is from mean sea level to h_2. Now returning to Equation 4.132, the optical density from infinity to P_2 is:

$$\alpha_{P_2} = \sec(\theta) \cdot \left[\sum_{j=N}^{i+1} \alpha_j + \Delta\alpha_i \right]$$

(4.135)

For the trivial case where the path is absolutely horizontal, then we can no longer integrate vertically from infinity (the $\sec(\theta)$ blows up). Equation 4.131 immediately reduces down to $\alpha = l \cdot \sigma_h$, where l is the distance between the two end points and σ_h is the constant extinction coefficient at altitude. To avoid the singularity near horizontal paths, we can use the average extinction coefficient along a *slightly* slanted path $\alpha = l \cdot (\sigma_{h_1} + \sigma_{h_2})/2$.

We now have enough to cast the above equations into a usable point-to-point atmospheric attenuation model. The code starts by importing a table of altitudes and extinction coefficients $\{h_i, \sigma_i\}$, one table for each laser wavelength. The first altitude is checked to make sure it starts at zero, if not then the code must linearly extend the extinction coefficients to zero height by adding a new first point. The last extinction coefficient is checked to make sure it is zero, if not then again the code must linearly extend the table by adding a new last point. Each air mass slab is indexed by the upper bounding altitude index. So the first slab extending from altitude h_1 to h_2 is given the index 2. Zero altitude is reserved for index 1 only. The slab index i is determined in practice by first finding the two bounding altitudes of the point in question using a binary search through the altitude table, and then taking the larger integer. If the point is above the highest

altitude then there is no atmospheric attenuation (i.e., α_p in Equation 4.135 is zero). If it is at sea level, then C_1 is the full integrated optical density.

The code then loops over each air mass slab and calculates the integral according to Equation 4.133, but without the $\sec(\theta)$ term (its effect is included later). To be useful we start at the highest altitude and work downward (backwards) and keep a running sum. If there are N altitudes in the table, then we recursively build a table of values C_i:

$$C_{i-1} = C_i + \frac{(\sigma_i + \sigma_{i-1}) \cdot (h_i - h_{i-1})}{2} \qquad N \ge C_i \ge 1 \qquad (4.136)$$

so on down to the first air mass slab just above mean sea level. We know that the top value C_N is zero (i.e., $C_N = 0$) because we have not traversed a full interval of atmosphere yet.

Next we need to be able to evaluate the partial integral in Equation 4.134. This contribution to the total optical density is an expansion in the partial distance from the slab's upper bounding altitude and the current altitude:

$$\Delta\alpha_i = \frac{\sigma_{i-1} - \sigma_i}{2 \cdot (h_i - h_{i-1})} \cdot (h_i - h)^2 + \sigma_i \cdot (h_i - h) = A_i \cdot (h_i - h)^2 + B_i \cdot (h_i - h) \qquad (4.137)$$

where of course:

$$B_i = \sigma_i$$
$$A_i = \frac{\sigma_{i-1} - \sigma_i}{2 \cdot (h_i - h_{i-1})} \qquad (4.138)$$

We know that $B_N = 0$ by definition, and $A_1 = B_1 = 0$.

The optical density from space to an altitude h along a vertical path then is a simple quadratic in the partial distance $\Delta h = h_i - h$:

$$\alpha(h) = A_i \cdot (h_i - h)^2 + B_i \cdot (h_i - h) + C_i \qquad (4.139)$$

An initialization table of $\{h_i, A_i, B_i, C_i\}$ is built using Equations 4.137 and 4.139. The integrated attenuation to an altitude h is then calculated via Equation 4.140. The net atmospheric attenuation along a slant path_length is:

$$\eta = \left| [\alpha(h_1) - \alpha(h_2)] \cdot \frac{path\_length}{h_1 - h_2} \right| \qquad (4.140)$$

Notice the $\sec(\theta) = path\_length/(h_1 - h_2)$ term is now explicitly included. We take the absolute value because the physical quantity η must be positive.

If the two end points of the propagation path are at nearly the same height, then the attenuation is approximately the mean extinction coefficient times the length of the path:

$$\eta \approx \frac{\sigma(h_1) + \sigma(h_2)}{2} \cdot path\_length \qquad (4.141)$$

The extinction coefficients $\sigma(h_i)$ must be interpolated at the respective altitudes using Equation 4.129. As we shall see, this approximation only holds for relatively short path lengths.

Then using Equations 4.130 and 4.131, the net atmospheric transmission is:

$$\tau_{atm} = e^{-\eta} \qquad (4.142)$$

So far we have assumed the earth's atmosphere is a simple plane parallel set of stacked air mass slabs. In reality for long horizontal paths the earth's curvature becomes important. For instance, two end points of a propagation path could be at exactly the same altitude, but be so far apart that the beam passes through much lower and denser atmosphere. This would completely invalidate the approximation in Equation 4.142.

In addition since the earth's atmosphere is relatively thin, terrain elevation changes are very important. For instance, at sea level the atmospheric transmission of a 1.315 μm laser to space is about 75% . At 5,000 feet it's more like 92%.

Consider a light propagation path between two points in the atmosphere shown in Figure 4.25.
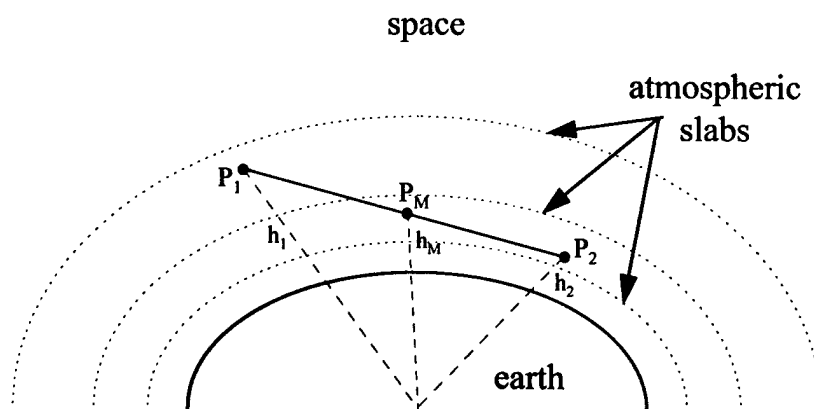
Figure 4.25. Earth with Concentric Atmospheric Air Mass Slabs
and a Propagation Path from P₁ to P₂

Even in this drawing there really is little atmospheric curvature between the two ends points. Because there is no analytic way to integrate the optical density between two points when curvature is present, we will need to approximate the linear path iteratively by subdividing it into shorter steps. Clearly as either end point moves toward the other, the earth curvature effects

diminish and the analytic approach developed above becomes exact. In fact for vertical paths it is exact (to within the approximations already made).

The simplest approach here is to first estimate the path integrals between the starting and end points using the above algorithm (i.e., assume the earth is flat). Then the interval between the two end points is divided into two shorter half spans. The optical density is again evaluated separately for each half span. If the sum of the two is reasonably equal to the whole span then the calculation is complete and the sum is used for the atmospheric transmission. If the sum of the half spans is significantly different from the whole span then the step size is too long.

The algorithm then must recursively iterate on each half span as it did with the whole span, dividing the step size in half until there is good correspondence between the half steps and the whole step for a particular sub-span. The net optical density is the sum of the values for each sub-span. Experience with this approach has been that only for long paths that graze the earth's surface does the number of iterations get large (50 or more), and these are very rare occurrences. Paths that are vertical, or pass through the upper atmosphere require only single iterations.

It is easy to subdivide an interval using the parametric equation of a ray. For example if the end points are separated by a distance d, and the direction cosines from a point P_1 to P_2 are $\hat{r}$, then:

$$\vec{P}_2 = \vec{P}_1 + d \cdot \hat{r} \tag{4.143}$$

where the carrot symbol means a unit vector, and $\hat{r} = \hat{i}l + \hat{j}m + \hat{k}n$. The midpoint P_M of the interval is at:

$$\vec{P}_M = \vec{P}_1 + \frac{d \cdot \hat{r}}{2} \tag{4.144}$$

Using Equation 4.144, one can easily find the midpoint position vector. Determining the altitude above mean sea level is a little more involved.

As we have discussed earlier, the earth is almost spherical. It is an oblate spheroid (ellipsoid) with flattened poles. The earth's oblateness is quite small, but it is easy to correctly model the earth as an ellipsoid. We have already stated that all geometry should be calculate in ECR coordinates. Thus, the points P_i, above in Equations 4.143 and 4.144, should all be vectors in ECR coordinates. The altitude of a point above an elliptical earth was discussed earlier in Equations 2.9 and 2.10. This is the altitude above mean sea level, not the altitude above the hard earth with terrain elevation variations. The atmospheric extinction coefficients are assumed to be constant with altitude (i.e., conformal) above mean sea level.

When one needs the atmospheric transmission between two known points (e.g., between the laser source and the target), then the two end points are input to the above algorithm. But, when one of the end points is not known, then we need to first solve for it. This occurs, for example, when a reflected ray is stopped by the earth's surface. We need to solve for the ray end point. If the ray intercepts the ocean (i.e., the earth ellipsoid), then one could consider a simple ray intercept

calculation. If the ray strikes land, then we will need to use an iterative algorithm to find the end point.

Figure 4.26 shows a downward propagating ray that is stopped by the earth, some distance above the mean sea level by local terrain.
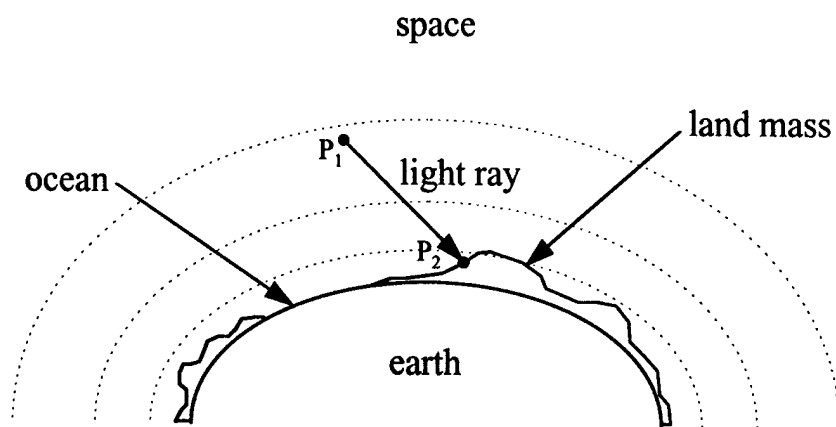
Figure 4.26. Terrain Elevations Elevate Ray Intercepts Above the Mean Sea Level

The terrain variations in Figure 4.26 are greatly exaggerated for clarity. First the ray is intercepted with the earth using the standard ray trace equations for an ellipsoid, at a point A in Figure 4.27. Then after converting the sea level intercepts into geodetic latitude and longitude, the local terrain elevation is found by bi-linear interpolation of the local terrain elevation database. Knowing the cosine of the incident ray with the local zenith vector (i.e., the surface normal), the ray is backed up until its elevation is about equal to the local elevation, which is point B in Figure 4.27. The new elevation is determined along with the new local terrain elevation at point B. The difference is used to move the ray again to point C, where the process is reiterated again. Finally the ray stops adjusting its position significantly at point D, the best approximate intercept with the earth.
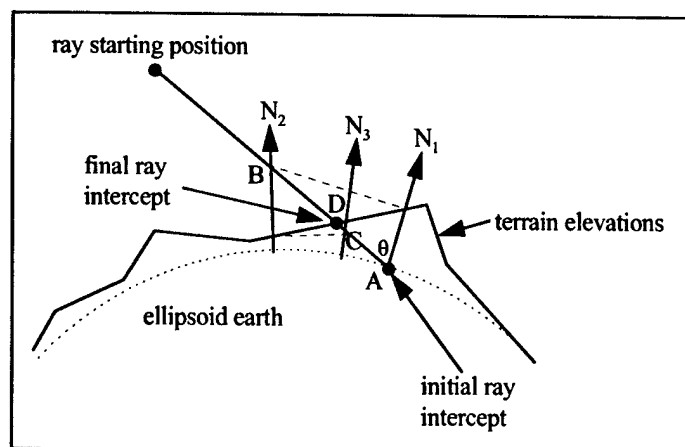
Figure 4.27. Iterative Ray-Earth Intercept Geometry Example

The final distance to the point D is the range to the ground, and the altitude is used to calculate the atmospheric attenuation.

For the 1.315 μm laser wavelength, the ABL SPO has used the total (molecular + aerosol) extinction coefficients plotted in Figure 4.28 [1].
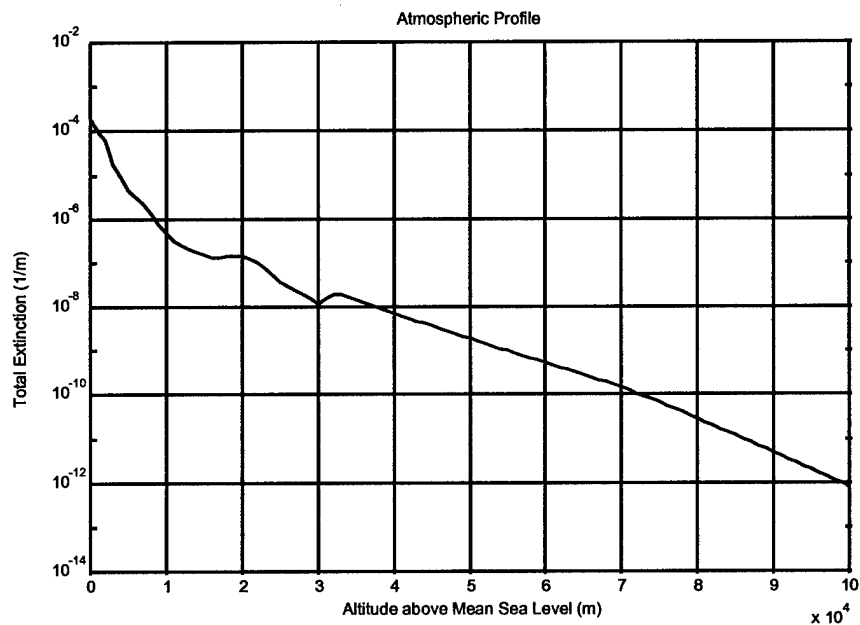
Figure 4.28. ABL Extinction Coefficients for 1.315 μm Wavelength

The net atmospheric transmission from a point located 15 km above the ocean surface to a point either in space or on the ground is plotted in Figure 4.29. Actually a fan of rays was generated and the transmission calculated to where either the ray exits the atmosphere to space, or when it intercepts the earth's surface.
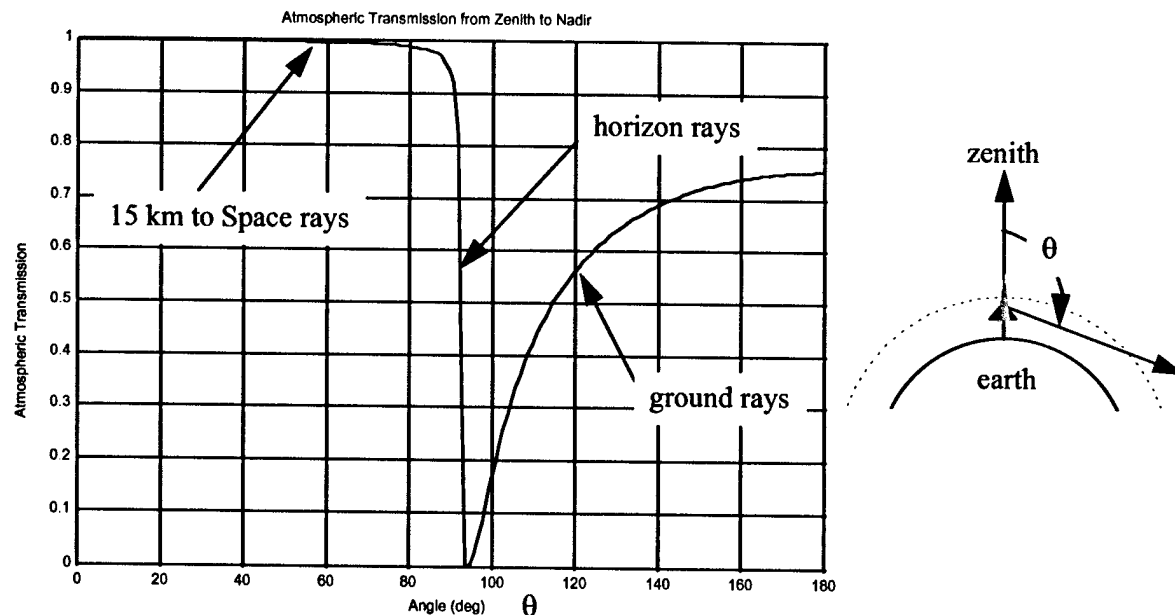
Figure 4.29. Atmospheric Transmission for a Fan of Rays Originating 15 Km Above the Ocean

As one would expect, the transmission plummets when the rays pass near the earth's horizon (i.e., around 86 degrees from zenith in the figure). The transmission to space is almost unity for upwelling rays. For ground rays the transmission increases with angle because the path length decreases. This would therefore be a typical transmission curve for a missile target.

5.0 Observer Modeling

The ultimate goal of the LRST code is to determine if and where scattered light hazards exist. Fortunately, most of space is not occupied by people or optical sensors that might suffer damage. This is not to say that a person or sensor might not exist virtually anywhere in space. This was the primary reason we were forced to develop the mechanics to calculate intensities at any point in space (except maybe for points very close to the target object). Observer modeling places observers throughout the scene at locations that we might expect to find people or sensors. We will essentially instrument the scene with lots of localized point observers, that will monitor the reflected intensities as a function of time. For each observer, its intensity time history profile will then be evaluated against damage criteria to determine where potential hazards exist. In this section we discuss how the scene is instrumented with observers, and how the intensity time history is developed for each observer.

For eye hazard analysis, there are four important kinds of observers. The first and most obvious would be *ground* observers, or those people standing anywhere on the earth surface (consistent with the terrain). This includes the ocean surface because they could be on board boats and ships, for instance. There are two forms of obstruction for ground observers. Clearly not everyone on the planet can see the missile body during a test illumination. Beyond the target horizon, there is no further need to instrument with ground observers. We therefore need to determine the location of the highest point of the target trajectory (that will be illuminated) to find that horizon circle. The other way people can be shielded from scattered light is by terrain obstructions, like mountains. While this certainly is possible, for the higher altitude missile trajectories (say above 10 km where they break the clouds) the fraction of the local earth's surface that remains shielded by terrain is usually very low (just a few percent). For the most conservative analysis, we therefore assume no terrain obstruction for any ground observer. We will need to instrument the earth surface below the target with a mesh or array of ground observers, no two being more than ½ degree apart as seen from the target. This was our basic resolution requirement stated earlier.

The second type of observer comes from a requirement to know the intensities around the military base from which a test missile might be launched. We already have assumed that the area within the base boundary has been instrumented with ground observers. We will next build a virtual fence around the base periphery that extends up to about 80,000 feet and instrument it with observers. While people are not likely to occupy such a surface, it is important nevertheless to monitor intensities and hazards on this fence.

The third type of observer recognizes that a fair amount of air-space is traveled by commercial airplanes throughout the day. We will assume airliners stay within prescribed air-lanes or air-corridors, such that we can build a bounding surface and instrument it with observers.

The fourth type of observer comes from test aircraft possibly involved in the test operations, which might actually fly fairly close to the target object. The observers are flight crew such as pilots that might view the target during the test illumination. In this case we do not have an arial mesh to build, but rather single dynamic points.

Finally, as a diagnostic, we will also include singular static point observers that can monitor specific points in space.

We will discuss shortly how the various observer classes are meshed, but for now assume that has been done. In Figure 5.1, we show how the dynamics of the laser source (e.g., the ABL), the missile flight dynamics, and the observer motions all combine to make the received intensity time history a complicated dynamical process. All three motions need to be taken into account to properly determine the correct intensity dynamics. To reduce that complexity some we work in a fixed earth frame (i.e., the ECR) since most of the observer arrays all have fixed observers. This leaves the motion of the laser source, and the missile flight dynamics. For each observer in the scene, we will first determine the effective line of sight dynamics for that observer as seen in the cross section sphere, interpolate the cross section across the sphere, scale with laser irradiance, range and atmospheric attenuation to develop a complete time history of the received intensity. While an observer point might be a member of one of the observer types discussed above, it actually exists in isolation of the others. We assume no obstructions for example. But, while each observer really does not know about its neighbors, it is placed such that the intervening regions between observers is assumed covered with an infinite number of observers to complete the coverage. By not actually instrumenting the scene with an infinite number of observers, we therefore are making the approximation that if we had, the intensity time histories would not be very different from those developed for the observer points that are implemented. This is valid provided the target is sufficiently distant from each observer, and the angular separation of neighboring observers is always within the ½ degree requirement. This approximation is usually met just by the nature of missile flight paths. The one possible difficulty might occur if the missile should fly through one of the mesh observers, such as the base boundary. Even here this is probably not an issue since people are not really expected to exist on the boundary fence. If a hazard should occur at any one observer point, it is assumed the hazard probably exists in the area covered between that observer point and its nearest neighbors. In practical terms a hazard polygon can be constructed and displayed in this area.

The ground observers are placed as a non-uniform mesh of points in concentric circles, starting directly below the highest point of the target trajectory. Each type of missile target has a particular engagement window based on its flight characteristics. For instance, missiles must be illuminated before launch motor burn-out, and sometime after it breaks above the clouds. The simulation should define the point in time where the lasers (especially the high energy laser) are turned off. This should also be the highest point in its trajectory where we determine the effective horizon.

The ground point directly below this point defines the pole of a spherical polar coordinate system. We will next construct concentric rings with decreasing latitude down from this pole. Enough observers will be generated to ensure the proper observer separation around the rings. The center rings are smaller, and hence will have fewer observers assigned to them. The outer rings near the visible horizon are spaced with greater polar angles such that the angular separation between the outer ring and next inner ring is again just ½ degree – as seen by the target. This method minimizes the number of observers while maintaining the proper resolution in any direction out to the horizon. Notice that this array scheme is not coincident with a rectangular world display. The observers are placed in altitude according to the terrain database.

Putting the observers at their local altitude could be important since high altitude points suffer less atmospheric extinction losses and shorter ranges.

The base boundary observers are placed on a fairly uniform mesh of points on a fence around the base. First the (closed) base boundary is defined by a (latitude,longitude) table read from a data file. This list most likely is a little coarse for what we want. We loop through the list of bounding points and look for angular separations. First, between any two bounding points we check the angular separation between them as seen from the target high point. If they are separated by more than the ½ degree, then we linearly interpolate new points around boundary in (lat,lon). We are given a required altitude the fence must extend up to. Then from the target high point, we find the angular separation between the bottom ground point of each fence post to this required altitude. The number of observers required vertically is the ratio of the angular height to the ½ degree resolution. Finally, using the maximum number of required observers for any fence post, we proceed from post to post and instrument each with observers.

The air corridor observers are placed on a fairly uniform mesh of points on an inverted u-shaped tunnel along the air corridor path. First the (non-closed) air corridor path is defined by a (latitude,longitude) table read from a data file. We loop through the list of points and determine how many observers must be used around the U shape and along the flight path to maintain the ½ degree resolution as seen from the target. Then along each segment of the corridor three planar faces are meshed with a rectilinear array of observers.

As the laser source moves, or the target moves, or the observer moves during a time step, the reflected intensities will change according to the changing geometry. Recall that we made the assumption that between time steps (perhaps on the scale of one second apart), the reflected cross section pattern would not vary significantly. This means that either the various players in the scene are very distant to the target, or the chosen time step is small enough to ensure this primary assumption holds. For ABL scenarios this seems quite easy to satisfy because of the large distances between the ABL and target, and any observers and the target. Now, one of the early concerns was the possibility of a sharp glint sweeping past an observer, within a single time step, such that the bright (and perhaps damaging) intensity spot would not be "seen" by an observer. This is a classic case of aliasing that we do not want to occur under any circumstances. By mapping the reflected intensities (i.e., cross sections) onto the surrounding cross section sphere, we assured that at least we have not missed their existence somewhere in the problem solution. The question is, can we arrive at an algorithm that will get each observer to properly sweep over this spot between time steps?

There are four types of motion that can change the intensity time-history an observer will see during a single time step. Figure 5.1 shows the combined effects of laser source motion, target motion, target rotation, and observer motion.
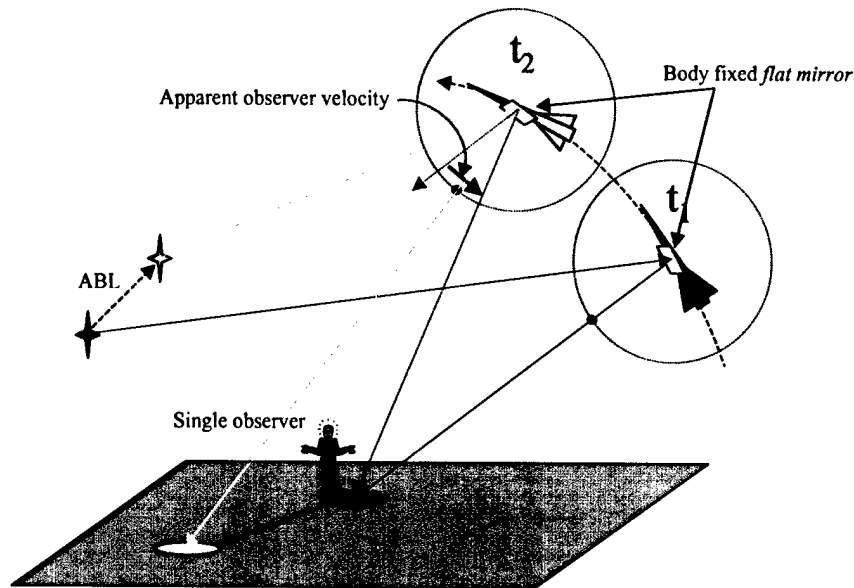
Figure 5.1. Scene Dynamics Induced Spot Motions

Start with a static (i.e., frozen in time) situation, as shown in Figure 5.1, and allow just the observer to move during a single time step. In Figure 5.1 the observer moves from the red dot to the blue, but neither the missile target nor the ABL laser move in time. This figure is drawn again in a single plane in Figure 5.2. To develop an intensity time-history over one time step for this case is easy. In direction cosine space (how we address pixels on the cross section sphere), we determine at time t_1 the LOS (unit vector) from the target origin to the observer $\hat{O}_1$, and then determine the LOS at time t_2 to be $\hat{O}_2$. If we difference these two vectors we get the difference vector $\vec{\Delta} = \hat{O}_2 - \hat{O}_1$.

At this point we make our second important assumption. We assume that during a single time step all motions can be approximated by simple linear motions of vectors between their end points. For the case where only the observer moves, we can read the cross sections for small time steps across the cross section sphere: $\chi_i = \chi(LOS) = \chi(\hat{O}_1 + i \cdot \vec{\Delta})$, where i stands for one point in time, and the vector argument should be re-normalized. If the angular separation is large compared to the resolution of the cross section sphere (i.e., ½ degree), then we would clearly want to sub-divide time into many minor time steps to make sure we visit every pixel between the end points (blue dots in Figure 5.2). As long as we do not step over a pixel, we will never miss even the smallest bright glint. The issue then is how fine of minor step do we take, or do we make the number of minor steps adaptive depending on the local geometry? Note also that as we traverse the sphere, the cross section data is held fixed in the Golden Sphere frame, at least during the time step interval. For subsequent time steps, the nearest appropriate Golden Sphere is used.
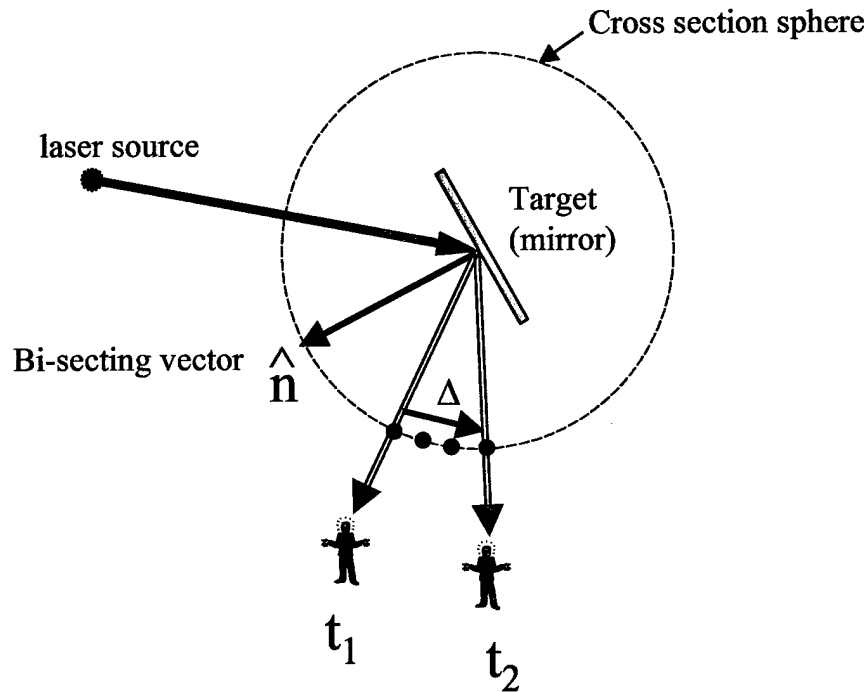
Figure 5.2. Observer Moves While Everything Else Remains Fixed

When the observers are ground observers, they do not move (at least not in the earth centered relative frame).

Next we hold everything static except the laser source. In Figure 5.3 we show the laser moving upward. Now we make a third important assumption, or approximation. As the laser LOS moves over small angles, the reflected cross section distribution moves linearly, at least on a local scale. In the figure we made the target a simple planar mirror. As the laser moves from position t_1 to t_2, the reflected LOS moves from R_1 to R_2.
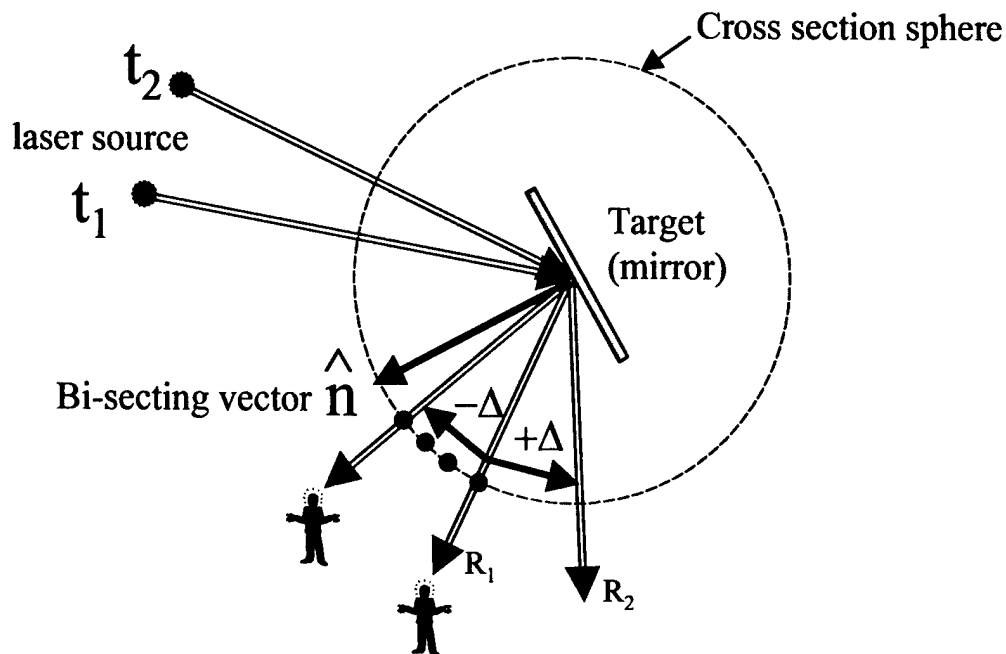
Figure 5.3. Laser Source Moves While Everything Else Remains Fixed

Thus if at t_1 an observer were located at position R_1, then at time t_2 the spot would no longer fall upon him because it had moved on to R_2. But, in the frame of the Golden Sphere, the cross section distribution does not move. So, in effect we must move the LOS to the observer in the opposite direction. Thus, if the change in the reflected LOS is $+\Delta$, then we must traverse the observer LOS along $-\Delta$.

Next we hold everything static except the target. If the target translates but does not rotate, then from the frame of the Golden Sphere both the laser source and observer might appear to move. In addition to translation, if we allow the target to rotate then again both the laser and observer will appear to move. Basically there are two effects, a moving observer and a moving spot, which is equivalent to the observer moving in the opposite direction. Using the above methods we can approximate how to traverse the cross section sphere. Figure 5.4 shows this basic algorithm.

```
┌─────────────────────────────────────────────────┐
│   Find LOS vectors to laser source at t₁ and t₂   │
└─────────────────────────────────────────────────┘
                        │
                        ▼
┌─────────────────────────────────────────────────┐
│    Find LOS vectors to observer at t₁ and t₂      │
└─────────────────────────────────────────────────┘
                        │
                        ▼
┌─────────────────────────────────────────────────┐
│           Find bi-secting vector at t₁            │
└─────────────────────────────────────────────────┘
                        │
                        ▼
┌─────────────────────────────────────────────────┐
│       Find evolution of bi-secting vector at t₂   │
└─────────────────────────────────────────────────┘
                        │
                        ▼
┌─────────────────────────────────────────────────┐
│               Reflect laser ray at t₂             │
└─────────────────────────────────────────────────┘
                        │
                        ▼
┌─────────────────────────────────────────────────┐
│     Subtract spot motion from observer motion     │
└─────────────────────────────────────────────────┘
                        │
                        ▼
┌─────────────────────────────────────────────────┐
│    Interpolate cross sections between end points  │
└─────────────────────────────────────────────────┘
```
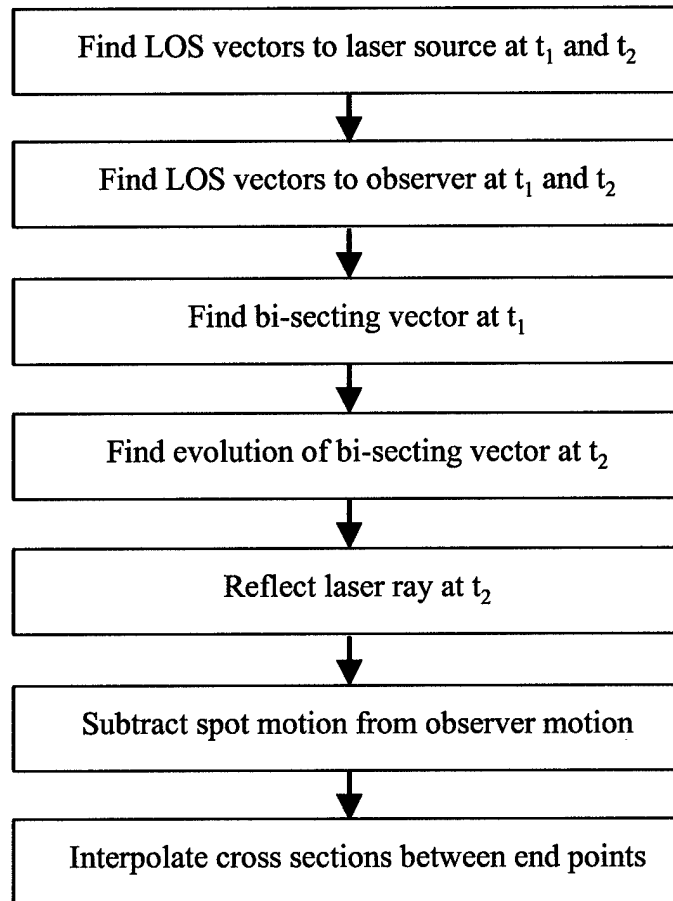
Figure 5.4. Basic Cross Section Sphere Traversal Algorithm

In the ECR frame we find the positions of the laser source at times t_1 and t_2. In the ECR frame we also find the positions of the Observer. We then find the bisecting vector $\hat{H}_1$ between the laser LOS and the observer LOS at time t_1. Again, consider the target as having a flat mirror whose normal is aligned with this initial bisecting vector direction, such that a ray starting at the laser will reflect exactly to the observer at time t_1. To find the effective spot motion, first transform the bisecting vector $\hat{H}_1$ into the body frame of the target, get the world to target transform for time t_2, and back transform to the ECR frame to get $\hat{H}_2$. Next use $\hat{H}_2$ to reflect the laser LOS vector at time t_2. The spot motion is the difference between the reflected direction and the original observer direction. Finally we subtract the spot motion from the true observer motion to get the total apparent motion of the observer relative to the fixed cross section distribution in the Golden Sphere frame. Knowing the initial observer direction and the final apparent observer direction, we subdivide the difference vector and interpolate the cross section sphere to get a series of intermediate cross sections. Again, the accuracy of this approach is dependent on our assumptions of small angular changes for the observer, the laser source, and the target dynamics.

This algorithm is repeated for each time step while the laser is on (i.e., between the beginning and ending times where we have precalculated the cross section spheres). The resulting time

history curve is thus generated for each observer. As mentioned above we could sub-divide the time step interval by either a constant delta, or by a changing amount determined by the effective spot motion dynamics. The former is easier to implement and has the advantage that the final time history array has a uniform time step, for all observers, and all times.

So far we have just discussed interpolating cross section values. In fact we do not want cross section, but rather an intensity time history profile at the observer. At each major time step we calculate the ranges R_1 and R_2 from target to observer, and use the smaller value. If the smaller range to the target is less than the distance to the horizon, then we continue (i.e., the observe is not obstructed by earth curvature). At each time step we calculate the peak irradiance I_0 for the beam. We then calculate the atmospheric transmission τ_{atm} from target to observer, using the lesser range. Finally we scale the cross section values to get intensities:

$$I_{observer}(i) = \frac{I_o \cdot \chi(i) \cdot \tau_{atm}}{R^2} \qquad (\text{w/m}^2) \qquad (5.1)$$

where the i stands for the ith minor time step.

6.0 Hazard Modeling

Once we have a time-history of intensity for an observer we can process it against some vulnerability model. For eye damage we use the ANSI Standard Z136. This model appears simple enough, but in fact is open to a bit of interpretation. Figure 6.1 shows a family of curves for eye damage levels as a function of laser wavelength, from the UV to the mid infrared. The exposure is assumed to be a uniform intensity for the intervals noted in the figure. The total flux the eye can tolerate is given by the ordinate. For long exposures (the top curve), the eye can tolerate more total received flux because the irradiance is so much lower. The curves are very non-linear with wavelength, as the figure shows.

Another way to plot the damage curves is to first select a wavelength, then plot the allowable radiant exposure (J/cm^2) as a function of the exposure interval (again, assuming a constant irradiance at the eye). Figure 6.2 is the maximum permissible radiant exposure curve for a 1.31 μm laser wavelength. If we assume the irradiance has a simple flat top pulse shape, then we can divide each value in the curve by the exposure interval to get the equivalent radiant intensity curve shown in Figure 6.3.
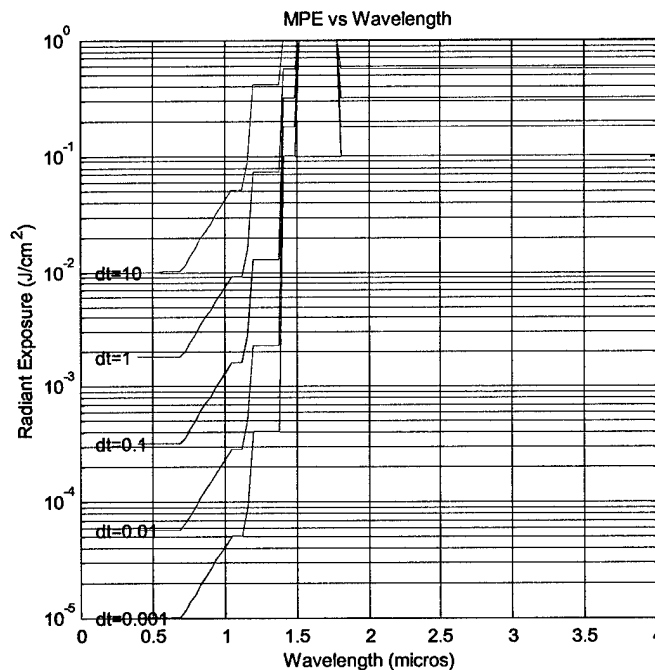
Figure 6.1. ANSI Z136 Curves as a Function of Wavelength
*for Various Constant Exposure Intervals*

This curve is perfectly valid as long as the irradiance is constant during the pulse on interval. If, however, the irradiance is not a simple pulse, but rather a complicated time-history function, then this figure is not necessarily valid.
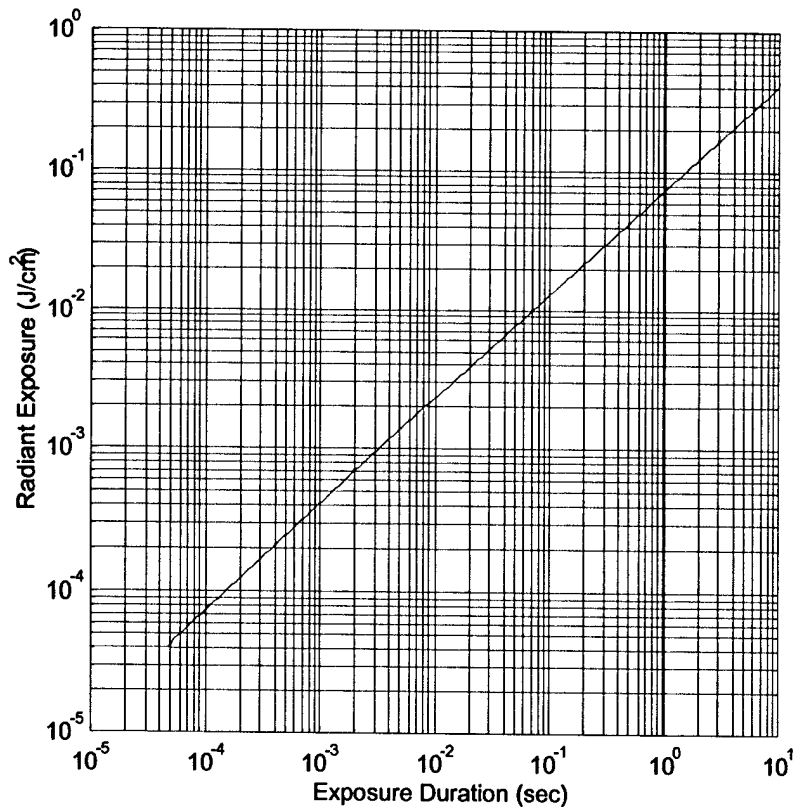
*Figure 6.2. Maximum Permissible Radiant Exposure Curve for a 1.31 μm Laser*

From Figure 6.3 it should be clear that the eye can certainly tolerate higher instantaneous pulsed intensities for the shorter exposure intervals. But, from Figure 6.3 we see that while the shorter interval exposures can have higher peak intensities, the total energy captured by the eye is less for the shorter pulse lengths.

Now, we know the ABL high energy engagement interval might be somewhere greater than 1 second, and probably more than 10 seconds long. The lasing time will depend on system performance, power output, range, target aspect angles, atmospheric transmission, and target hardness. We must therefore assume that the laser will probably be on for at least 10 seconds. But, we must set the maximum exposure interval to 10 seconds because there is a natural tendency for humans to not dwell on a specific point for longer than this. We therefore are lead to the immediate conclusion, based on the assumed 10 second exposure interval, that if the irradiance at the eye never goes above about 0.04 W/cm² (the exact value is calculated using the ANSI Z136 equations), there can be no hazard or eye damage, period. But, does this mean that if we get a short spike (say a millisecond or less) that exceeds the maximum allowable irradiance for a 10-second interval, we necessarily have a hazard? Obviously the answer is no. Clearly we can say that any short lived spike that exceeds the maximum allowable intensity for that interval is a hazard, but what about low intensity spikes above the 10 second threshold?
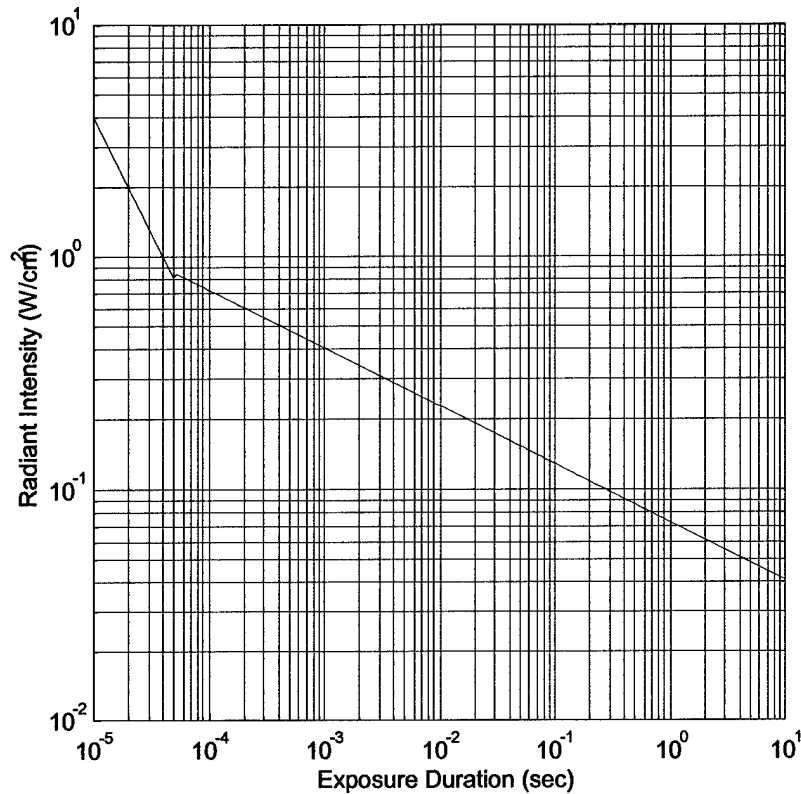
Figure 6.3. Maximum Permissible Radiant Intensity Curve for a 1.31 μm Laser

Before we discuss complicated intensity time-history analysis, let us define a second threshold. When we discussed how an intensity time history is developed, we needed to define a minor time step to ensure we never stepped completely over one of the cross section sphere pixels. If our basic simulation time step is one second, then the minor step might be on the order of 0.1 second. Therefore in Figure 6.4 we can define two critical thresholds and three regions to consider. First we can say that if no instantaneous intensity in a complicated time-history ever exceeds the 10-second MPE (about 0.04 w/cm^2), then there is <u>no</u> possible hazard for that observer (I_{min} in the figure). In Figure 6.4, this would be the green band. On the other hand, given a minor time step, if any instantaneous intensity value ever exceeds the MPE for that minor time step (0.14 w/cm^2 for a 0.1 second minor time step for example), then there is a hazard for that observer (I_{max} in the figure). In Figure 6.4 this would be the red band.
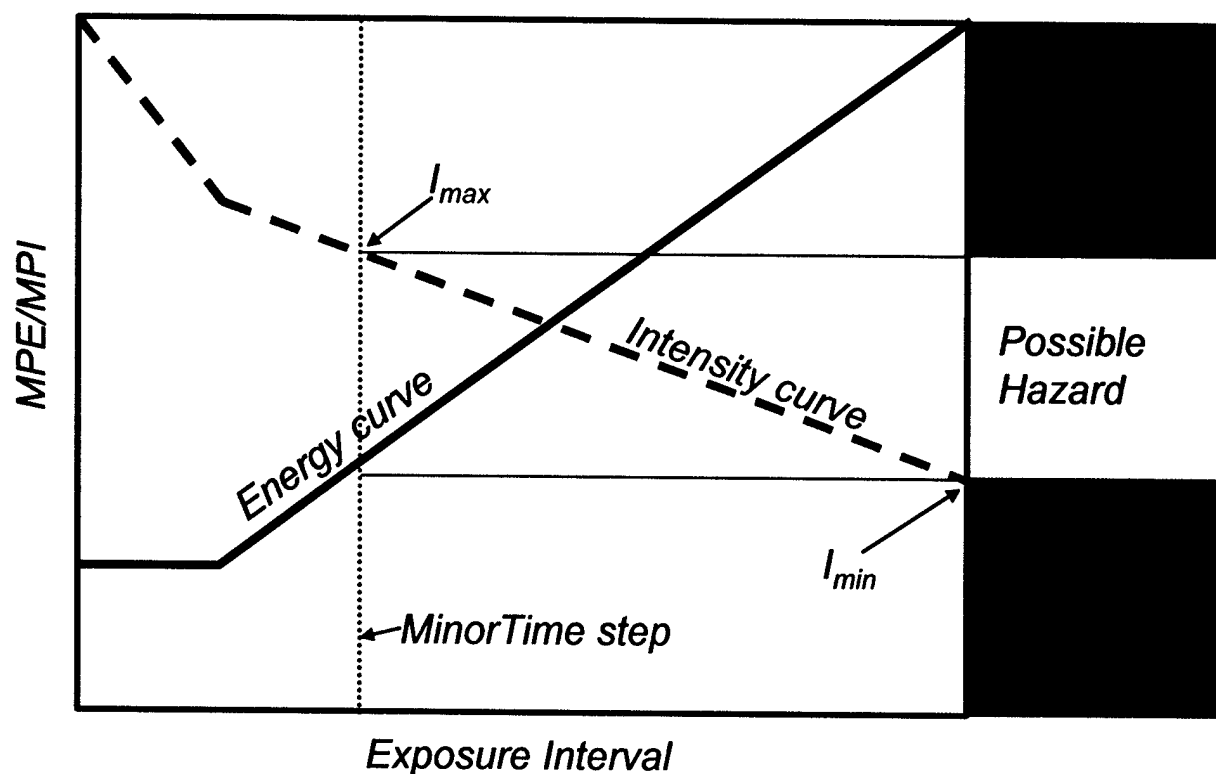
Figure 6.4. Intensity Thresholds and Hazard Bands

Is there a possibility of a rapid specular reflection sweeping past a ground observer in LRST? For instance, we might have a very shiny missile body (which is highly unlikely) that is moving at say 1 km/sec while being illuminated by a high power laser. Let us say the reflected intensity pattern is even a mere meter wide at the ground, and therefore moving at 1 km/sec. What minor step size should be use? Recall the cross section sphere is sampled with a ½ degree resolution. A half degree cone is about 130 meters at the ground (15 km missile altitude). It therefore takes about 0.13 seconds for this glint to pass from one pixel to the next on the Golden Sphere, which is about 1/10 of the typical time step. We now conclude that there are no millisecond type events to be expected in LRST.

For complex intensity waveforms where an occasional intensity value may exceed I_{min} but never I_{max}, we need to evaluate further if there is a hazard or not. This is the yellow band in Figure 6.4. The basic MPE curve of the ANSI Z136 basically limits how much laser flux can be accumulated by the eye within any given interval. If the eye accumulates too much flux too fast, damage might occur. Therefore we must take an intensity time-history curve and make sure that for no interval has the eye captured too much flux.

The algorithm we use to determine a hazard starts by checking the two intensity threshold limits discussed above. If at least one intensity value exceeds the 0.04 w/cm^2 threshold, but none exceeds the I_{max} threshold (depends on minor time step), then we must proceed with a more complex hazard evaluation algorithm. By this point we have already determine no single spike has crossed the hazard curve for the smallest time step. We proceed through the intensity time-

history curve two points at a time, adding the intensities together and multiplying by two minor time step intervals (simple Euler integration of the flux). The integration can be expressed as:

$$E_i(n) = \Delta t \cdot \sum_{j=i}^{i+n} I_j \qquad (j/cm^2) \qquad (6.1)$$

where i relates to the ith time sample, n is the number of intervals integrated, Δt is the minor time step, and I_j is the intensity at the jth time. Thus for n=2, we compare: $E_i(2) < MPE(2\Delta t)$ for each i. If none exceed that threshold, then we proceed with the integration for three time steps, and so forth until we have integrated the whole interval. Figure 6.5 attempts to show the idea graphically. The top curve shows two points being integrated. Different colors are different integrations. The second curve is the integration for three time steps (actually it is done for odd points too). The bottom curve is the last one done over the whole interval.
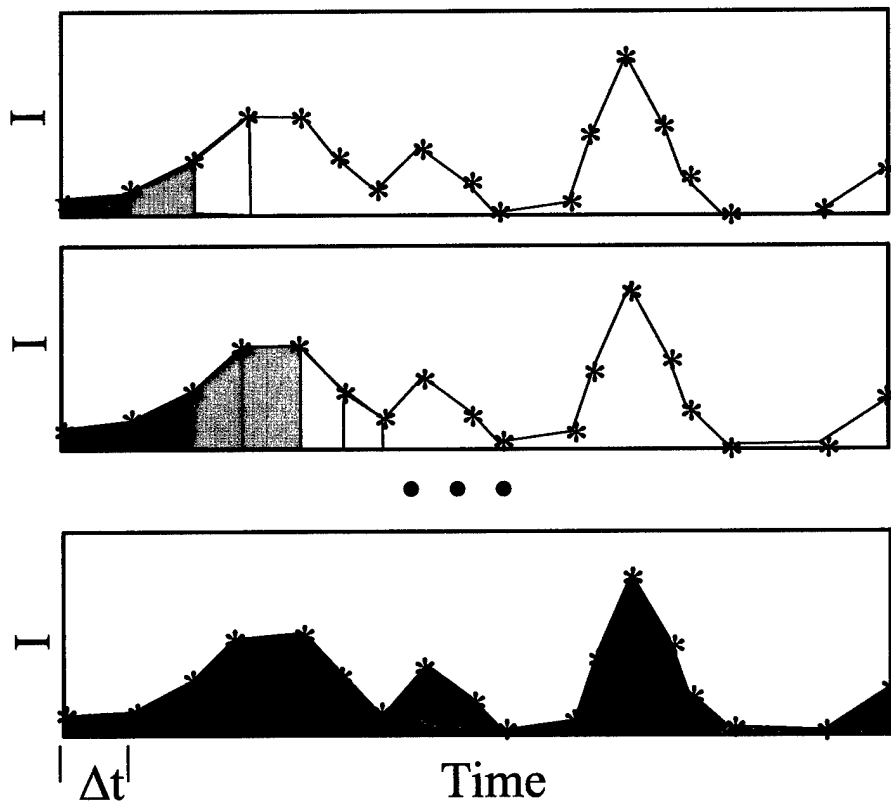
Figure 6.5. Intensity Time-history Integrations for Increasing Interval Lengths

This is a fairly complex algorithm, and takes a fair number of computations to complete one observer. It is unusual for the code to have to proceed through it, however, because for most observers the intensities never exceed the I_{min} threshold, or if there is a problem we tend to find intensities that exceed the I_{max} threshold too. From running test cases with simple waveforms we find that the margin gained by doing the complex processing is quite minimal. One might gain a

factor of two in instantaneous intensity for instance without incurring a hazard using this approach.

In Figure 6.6 we show why we do not really expect hazards for ABL. The intensity at an observer with no atmospheric losses is:

$$I_{observer} = \frac{\chi \cdot I_o}{R^2} \qquad \text{(w/cm}^2\text{)} \qquad (6.2)$$

We can invert this to give the required keep-out distance to remain at safe intensity levels:

$$R_{KeepOut} = \sqrt{\frac{\chi \cdot I_o}{I_{threshold}}} \qquad \text{(m)} \qquad (6.3)$$
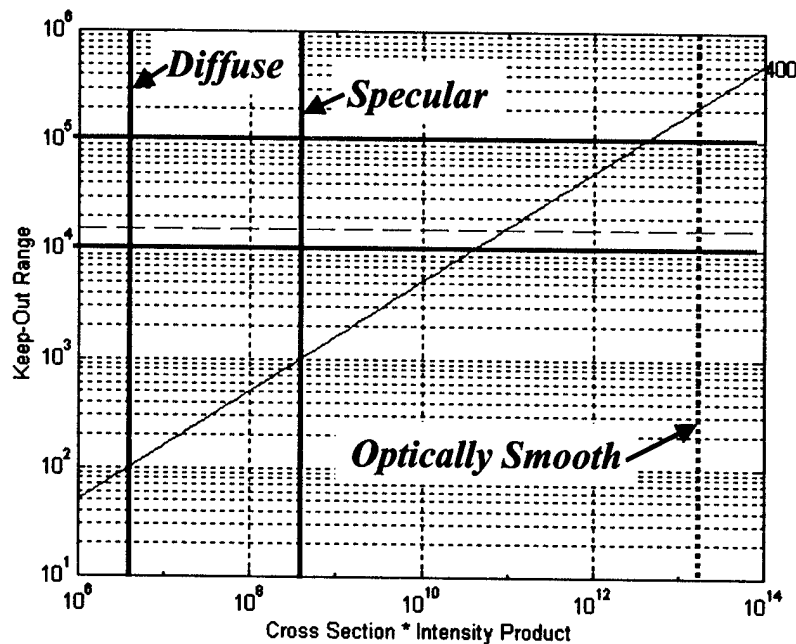
Figure 6.6. Expected Keep-out Ranges for ABL Scenarios

Given a cylindrical missile body shape and typical ABL high power irradiance values we can bracket the irradiance - cross section product from diffuse to fairly shiny BRDFs by the two vertical red lines in Figure 6.6. The blue line is the 0.04 w/cm^2 threshold level used in Equation 6.3. The bottom green line drawn horizontally is about the height of typical clouds. The upper green line is space, where we probably would not be able to engage missiles (or would they likely still be thrusting). The red dashed line is at 15 km altitude, which is about where we expect a missile to be after breaking cloud cover, and the minimum distance to any ground observers. For even a very shiny BRDF, we have at least an order of magnitude of range margin. Note the keep-out distance increases as the square root on the reflected cross section. On the other hand, if we

assume an optically smooth surface, then we are probably assured of ground hazards out to 200 km, which is half way to the horizon.

So far we have only considered eye hazards for people looking directly at the target missile during the engagement. There also exists the possibility that they could be looking through optics that either enhances the light collection (e.g., binoculars or a telescope), or attenuates the exposure (such as glasses designed for Laser Eye Protection (LEP)). In either case we want to simply multiply the intensity time history values by a scalar constant.

There are a couple of aspects we need to consider for doing pulsed laser radiometrics and eye-hazard analysis. First we need to consider how to add the time-dependence of the laser to the radiometric modeling. The end objective of all the LRST modeling is to resolve issues of vulnerability for either human eye damage or sensor damage. In either case, the fundamental mechanism is some kind of thermal damage, which is driven by energy, the integral of power. In the case of pulsed lasers, the output of the device is rapidly time-varying, i.e., pulsed, which adds temporal complexity to the otherwise slowly time-varying spatial distributions of reflected power off the target.

The pulse rate could be very fast, many kilo-hertz and above, in which case the effective irradiance would be equivalent to a continuous wave (CW) laser, and thus the pulsed aspect would become irrelevant. One would in this case want to treat the laser as a CW source and consider no further any details about its frequency or pulse shape. According to the ANSI standard, this occurs at 55 kHz for wavelengths 0.4 µm to 1.05 µm, and 20 kHz for wavelengths between 1.05 µm and 1.4 µm. How we model frequencies outside this wavelength range is unknown. Now, if the user defines a laser source with frequencies above these respectively, the user-interface of the LRST code should internally substitute a CW parameter set. The user could be informed a CW model is being used, but only for convenience.

For pulse repetition frequencies (PRF) below these respective rates, the ANSI standard demands a slightly different treatment of the eye vulnerability calculations relative to CW cases. We will assume that the temporal waveforms of the laser output have absolutely no impact on the spatial distribution of light propagating to the target (i.e., the wavefront quality) beyond the fact that the little photons are merely bunched up longitudinally. All the remaining mechanics for propagating the beam to the target and reflecting it to the observers is absolutely the same. We will not worry about detailed temporal effects such as doppler shifts, pulse broadening due to target depth variations, and so forth. We assume the pulse waveform at the laser is exactly the same as that at the observer, except the power levels are greatly reduced by all the propagation and scattering physics between laser and observer.

Clearly at the opposite extreme is very slow pulse rates, such as one pulse every ten seconds (we limit the integration time to ten seconds in any event), then the single pulse's width becomes the integration interval relative to the ANSI MPE curve. Again, if the PRF is at or below the inverse of the laser on-time (e.g., 0.1 Hz for a 10 second engagement), then the user interface should set up the scenario at a *single* static point in time, set the laser power to the pulse peak, and set the engagement duration to the pulse width.

For PRF's between these two extremes, we need to run LRST in a mode similar to using a CW laser and rely on the ANSI standard to help us define how to modify the MPE calculations.

The waveform of a laser can, in principle, be anything from a simple square wave pulse train to a very complicated set of multiple minor pulse shapes, none of which having a well defined peak or width, and certainly never a square shape. Nevertheless, we can usually derive an approximate repetition period for the waveform, some kind of effective width to the pulse, and its effective *peak* power value. The intent of the user input laser power parameter is always to be this *peak* value, not the average. Throughout the code we calculate peak values of spatial intensity, or laser irradiance, as if it were always CW.

Including the temporal pulsed laser effects into the hazard analysis is done only at the point where the intensity time-history for any observer is compared with the ANSI standard. The pulsed laser effects are captured solely in how the various MPE's are constructed. If the beam is CW then we basically use the MPE curve equations in the ANSI standard handbook, based on wavelength. If the laser is pulsed, then we first determine the number of pulses in an integration interval Δt:

$$n = \max(\ \Delta t \cdot PRF \ ,1.0) \tag{6.4}$$

Next we determine the MPE(t_{pulse},λ) for a single pulse width. The *adjusted* single pulse MPE$_{adjusted}$ is this MPE(t_{pulse},λ) times the number of pulses n divided by the fourth root of the number of pulses:

$$MPE_{adjusted} = MPE \ (t_{pulse} \ ,\lambda) \cdot n \ / \ n^{1/4} = MPE \ (t_{pulse} \ ,\lambda) \cdot n^{3/4} \tag{6.5}$$

This equation comes straight from the ANSI documentation. Next we calculate the MPE$_{average}$ for a continuous on pulse of the same interval:

$$MPE_{average} = MPE \ (\Delta t, \lambda) \tag{6.6}$$

Finally, we take the lesser of these two MPE values for our final conservative MPE:

$$MPE = \min(\ MPE_{average} \ , MPE_{adjusted} \ ) \tag{6.7}$$

This MPE is used to compare with the expected reflected intensity at an observer:

$$I_{observer} \ \cdot \Delta t \cdot DutyCycle \ < MPE \tag{6.8}$$

where the incident intensity is adjusted to be an effective *average* intensity, not the peak, by the duty cycle. Actually we rescale the MPE as:

$$I_{observer} \ \cdot \Delta t < \frac{MPE}{DutyCycle} \tag{6.9}$$

Right or wrong, we reasoned that the MPE is an integrated value, and so the intensity should be an average value. This might not be the most conservative method, but its what was implemented in the LRST code. All the radiometric calculations everywhere else in the code assumes a *peak* cw laser power value. It is only at the last eye-hazard evaluation that we look at the detailed temporal waveform of the laser power. So, for instance, a display of intensity on the ground would be for the *peak* instantaneous intensity, never the average.

7.0 References

1. Vallado, David, "Fundamentals of Astrodynamics and Applications," McGraw Hill, 1997.

2. Born, Max, and Wolf, Emil, "Principles of Optics," Pergamon Press, Sixth Edition, 1980.

3. Yariv, Amnon, "Optical Electronics," CBS College Publishing, Third Edition, 1985.

4. Crockett, G., "Bi-directional Reflectivity Distribution Function (BRDF) Modeling in LRST: Maxwell-Beard, Phong, and Gaussian Models, Contract F04701-98-D-0100, CDRL A004," January, 1999.

5. J. R. Maxwell, J. Beard, et al, *Bi-directional Reflectance Model Validation and Utilization*, ERIM Technical Report AFAL-TR-73-303, October 1973.

6. B. T. Phong, "Illumination for Computer-Generated Pictures," *Comm. ACM*, 18(6), 1975.

7. K. E. Torrance and E. M. Sparrow, "Theory for Off-Specular Reflection from Roughened Surfaces," *J. Opt. Soc. Amer.*, Vol. 57, No. 9.

8. *Non-conventional Exploitation Factors (NEF) Modeling*, ORD-257-96, 22 August 1996.

9. G. Crockett, "The Derivation of Analytic Optical Cross-Sections for a Sphere, Cylinder, Cone, and a Flat Plate," Contract F29601-95-C-0197, CDRL A005.

10. H. B. Holl, "Specular Reflection and Characteristics of Reflected Light," *J. Opt. Soc. Amer.*, May 1967, Volume 57, No. 5, p. 683.